

The minimum distance of classical and quantum turbo-codes

Mamdouh Abbara*

Jean-Pierre Tillich*

Abstract

We present a theory of quantum stabilizer turbo-encoders with unbounded minimum distance. This theory is presented under a framework common to both classical and quantum turbo-encoding theory. The main conditions to have an unbounded minimum distance are that the inner seed encoder has to be recursive, and either systematic or with a totally recursive truncated decoder. This last condition has been introduced in order to obtain a theory viable in the quantum stabilizer case, since it was known that in this case the inner seed encoder could not be recursive and systematic in the same time.

1 Introduction

Turbo-codes are a class of codes with very good properties. Provided that the outer encoder and the inner encoder are chosen according to certain requirements, namely, that the minimum distance d_c of the outer encoder is greater than or equal to 3, and the inner encoder is recursive and catastrophic, then the random turbo-encoders built by linking these encoders by the means of an intermediate random interleaver have a minimum distance which behaves asymptotically as $N^{\frac{d_c-2}{d_c}}$ where N is the size of the input; if the minimum distance of the outer encoder is equal to 2, then the minimum distance attains $\log N$ *at best* for a particular choice of the interleaver, whereas the minimum distance is bounded for a randomly chosen interleaver. On the other hand, if the inner encoder is moreover non-catastrophic, these turbo-codes show good decoding performances for the symmetric channel as well as for the erasure channel, under a regime near to the Shannon limit, and with an iterative decoding algorithm with linear complexity in N . The property of having an unbounded minimum distance is lost if the inner encoder is either non recursive or non systematic, and the linear iterative decoding algorithm becomes useless if the inner encoder is catastrophic, since the decoder can not extract any useful information about the local probability distribution on the state of each bit.

1.1 The classical argument and where it fails in the quantum case

Attempts [2] were made to design quantum stabilizer turbo-codes with unbounded minimum distance and good decoding performance. However, it was found in [2] that quantum inner convolutional encoders can not be recursive and non catastrophic in the same time, nor can they be, as a consequence, recursive and systematic. In this paper, we overcome the difficulty and introduce a different set of conditions under which quantum stabilizer turbo-encoders have an unbounded minimum distance. Our result is partially inspired by [1]. The counting

*INRIA, Equipe Secret, Domaine de Voluceau BP 105, F-78153 Le Chesnay cedex, France.

argument we adapt in this paper comes by [1] and has the following outline. First, upper bound the number $a_{out}(w)$ of possible inputs of the outer encoder for which the output has weight w , and the number $a_{in}(w, \leq d)$ of possible inputs of the inner encoder for which the output has weight $\leq d$. Using the randomness of the interleaver, this implies an upper bound on the number of possible inputs of the turbo-encoder such that the weight of the intermediate output (after applying the outer encoder) is w and the weight of the final output is $\leq d$. By summing this upper bound over all possible values of w one gets an upper bound on the number of possible inputs for which the output weight is less than d . Kahale and Urbanke show that if d is $O(N^{\frac{d_c-2}{d_c}})$, such a sum tends to 0 as N tends to infinity. A very crucial hypothesis in the counting argument which can not be used in the quantum case is that one can suppose that $w \leq d$ because the inner encoder is systematic. This is where a new hypothesis has to be made in the quantum case, in order to have an acceptable upper bound for the sum in the new regime where $w > d$. A possible new hypothesis we investigate in this paper is that the truncated decoder, ie the decoder where the ancillary positions are thrown, corresponding to the inner encoder is *totally recursive*.

1.2 Results obtained under the common framework

In order to highlight the similarities and the differences between the classical and the quantum stabilizer settings, we define a formal framework to which both settings correspond as a particular case. The main idea behind this framework is to consider encoders as operations which act on a group of errors \mathcal{P} (\mathbb{F}_2 or the factor Pauli group depending on the settings) rather on the real input states. Under this formalism, a turbo-encoder is the concatenation of three operations: an outer encoder $C_{out}^{\otimes N_{out}}$ equal to the blockwise repetition of a seed encoder C_{out} , a random interleaver, and an inner convolutional encoder $C_{inN_{in}}$. We generalize the concept of *recursive* and *systematic* encoders, and introduce another concept of *total recursiveness*. We also write the definition of the *distance* d_c of an encoder, and introduce the *degenerate distance* d_q . This distance is not different from d_c in the classical setting, but it has a very important role in the quantum case since d_q comes instead of d_c in the expression of the lower bound on the minimum distance of the turbo-encoder. Some concepts like the *detours* of a convolutional encoder are more delicate and need to be redefined to comply with the quantum case. We then show the following results. The first result is only useful in the quantum setting since this is where $|\mathcal{P}| > 2$. The second result is only useful in the classical setting, since in the quantum setting the conditions on the inner encoder are impossible to satisfy (see [2]). Actually, the second result corresponds to the classical result stated by Kahale and Urbanke in [1] and the proof hereby follows the unpublished proof of Kahale and Urbanke.

Theorem 1.1. *Consider a turbo-encoder T_N of size N , where the interleaver is randomly chosen according to a uniform distribution. Let d_c and d_q be respectively the distance and the degenerate distance of the outer seed encoder. If all the following conditions are realized:*

- $|\mathcal{P}| > 2$
- $d_q > 2$ (case 1) or $d_c > d_q = 2$ (case 2)
- *the inner seed encoder is recursive, and the associated truncated decoder is totally recursive*

then with probability going to 1 as $N \rightarrow \infty$, the distance of the turbo-encoder is greater than:

- case 1: N^α for all $\alpha < \frac{d_q-2}{d_q}$
- case 2: $\alpha \frac{\log N}{\log \log N}$ for all $\alpha < d_c - 2$

Theorem 1.2. *If all the following conditions are realized:*

- $d_q > 2$
- *the inner seed encoder is recursive and systematic*

then with probability going to 1 as $N \rightarrow \infty$, the distance of the turbo-encoder is greater than N^α for all $\alpha < \frac{d_q-2}{d_q}$

In the counting argument needed to prove these results, we will separate the sum into three partial sums. The first partial sum corresponds to the case where $w \leq d$ as in the classical case. The second and the third partial sums correspond to the regime where $w > d$, and are separated into the case where w is sublinear and the case where w is linear with respect to the input size. These three partial sums will be upper bounded by an expression tending to 0 when the input size tends to infinity, under a set of hypothesis proper to each of them. The article is organized as follows. In Section 2, we present the common framework. In Section 3, we focus on the definition and some properties of the inner encoder: we introduce the formalism of convolutional morphisms, define the notions of *recursiveness*, *systematicity* and *total recursiveness*, and define the *speed* η of a convolutional morphism. In Section 4, we establish two upper bounds on the weight distribution $a_{in}(w, \leq d)$ (and actually $a_{in}(w, d)$) of the inner convolutional encoder. The proof is delayed to Appendix A. In Section 5, we establish two upper bounds on the weight distribution $a_{out}(w)$ of the outer encoder. In Section 6 we state the upper bounds on the three partial sums needed to proof each of Theorem 1.1 and Theorem 1.2, and we assemble these bounds to prove Theorems 1.1 and 1.2. The proof of the partial sums is delayed to Appendix B and relies on the bounds established in Sections 4 and 5.

2 The common framework

Focusing on errors propagation under an encoding process, rather than on the code or the encoding process itself, enables to put linear classical codes and quantum stabilizer codes under one common framework, and to grasp distance properties of the underlying code. For an introduction to the theory of stabilizer codes, the reader can refer to [ref codes stabilisateurs]. Errors to consider in order to know the minimum distance of a code are bit flip errors for linear classical codes and Pauli errors for quantum stabilizer codes. In the first case this is the direct consequence of the definition of the minimum distance of a classical code; in the second case, this comes from the fact that a subspace is a quantum error correcting code for a set of errors if and only if it is a quantum error correcting code for a vector space basis of these errors. We will call classical and quantum setting, respectively, the model of encoding and error propagation for linear classical codes and for quantum stabilizer codes.

2.1 Encoding protocol

In the classical setting, a (n, k) code is a linear subset of 2^k elements of the set of n bit states \mathbb{F}_2^n . Such a code is the image of an encoding protocol from \mathbb{F}_2^k to \mathbb{F}_2^n , which first appends

$n - k$ ancillary bits set to 0 and then applies a \mathbb{F}_2 -group automorphism \mathcal{V} of \mathbb{F}_2^n . It is thus the set of all $\mathcal{V}(\psi, 0^{n-k})$ where ψ describes the set of all k -bits states.

In the quantum setting, a (n, k) code is a 2^k -dimensional \mathbb{C} -subspace of the space of n -qubits states $\mathcal{H}^{\otimes n}$, where $\mathcal{H} = \{\alpha|0\rangle + \beta|1\rangle, (\alpha, \beta) \in \mathbb{C}^2\}$ designates the one qubit space. It is the image of an encoding protocol from $\mathcal{H}^{\otimes k}$ to $\mathcal{H}^{\otimes n}$, consisting in the addition of $n - k$ ancillary qubits set to $|0\rangle$ followed by a \mathbb{C} -vector space automorphism \mathcal{V} of $\mathcal{H}^{\otimes n}$. Again it is the set of all $\mathcal{V}(\psi \otimes |0\rangle^{n-k})$ where ψ describes the set of all k -qubits states. \mathcal{V} is more precisely a Clifford transformation, so that it stabilizes the group of Pauli transformations by conjugation.

Putting the encoding in the form of this protocol has the advantage to identify $n - k$ positions which will carry all the information about the error given by the decoding process \mathcal{V}^{-1} .

2.2 Error propagation: the encoder

Suppose that in the classical setting, an n -bits state $(\psi, 0^{n-k})$ is subject to an error $E = E_1 \dots E_n \in \mathbb{F}_2^n$, where $E_i = 1$ if and only if a bit flip happens at position i . After applying \mathcal{V} , we get the state $\mathcal{V}(E + \psi) = \mathcal{V}(E) + \mathcal{V}(\psi)$, which means that the effect of \mathcal{V} on the set of errors \mathbb{F}_2 is the group automorphism \mathcal{V} itself.

In the quantum setting, suppose that an n -qubits state $(\psi \otimes |0\rangle^{n-k})$ is subject to a Pauli error $E \in G_n$. After applying \mathcal{V} , we get the state $\mathcal{V}E\psi = (\mathcal{V}E\mathcal{V}^{-1})\mathcal{V}\psi$, which means that the effect of \mathcal{V} is to map each element E of G_n into $\mathcal{V}E\mathcal{V}^{-1}$, which also belongs to G_n since \mathcal{V} is a Clifford transformation. The operation $E \rightarrow \mathcal{V}E\mathcal{V}^{-1}$ is a group automorphism of G_n . Elements of G_n contain a global phase in $\{1, i, -1, -i\}$ which can be omitted for E as well as for $\mathcal{V}E\mathcal{V}^{-1}$ without any loss of information; this corresponds to considering the quotient Pauli group $G_n/Z(G_n)$, equal to the n -fold cartesian product of the quotient Pauli group of 4 elements $G_1/Z(G_1) = \{I, X, Y, Z\}$. The induced action of the operation $E \rightarrow \mathcal{V}E\mathcal{V}^{-1}$ on the quotient Pauli group is, again, a group automorphism.

Thus it is possible to unify both settings by writing that \mathbb{F}_2 and $\{I, X, Y, Z\}$ constitute a *group of errors* \mathcal{P} with neutral element I . The group of errors also needs to verify a property with respect to the decoding step and stated in the upcoming definition of the common framework. We then say that the automorphism of \mathcal{P}^n engendered by the encoding process, together with the knowledge of k to separate the information carriers positions from the ancillary positions, constitute an $[[n, k]]$ *encoder* C . Let us now push the comparison between the two settings a little further, by underlining the concepts of *undetected* errors and *harmless* errors.

2.3 Types of errors and the distances d_c and d_q

Suppose now that a non trivial error affects the state obtained after the encoding operation. Since the $[[n, k]]$ encoder C is an automorphism, we can write this error in the form $C(E)$ with E a non trivial error. When the decoding operation \mathcal{V}^{-1} is performed, the state left is $E(\psi, 0^{n-k})$ or $E(\psi \otimes |0\rangle^{n-k})$ depending on the setting. In both settings, one sees that if the last $n - k$ positions are left unchanged by E , the error $C(E)$ is *undetected*. This happens in the classical setting if the last $n - k$ coordinates of E are equal to 0, and in the quantum setting if the last $n - k$ coordinates of E are in the set $\{I, Z\}$ of errors acting trivially on the qubit $|0\rangle$. In the case of an undetected error, we are left with a state in the form $(\psi', 0^{n-k})$

or $(\psi' \otimes |0\rangle^{n-k})$ depending on the setting; if $\psi' = \psi$, the error $C(E)$ is *harmless*, otherwise it is *harmful*. In both settings, the minimum distance of the code is the minimum Hamming weight of a harmful error. We simply call this value *distance* and write d_c , whereas the minimum Hamming weight $d_q \leq d_c$ of an undetected error will be called *degenerate distance*.

2.4 Definition of the framework

Definition 2.1. • A group of errors \mathcal{P} is a finite group with composition law $*$, neutral element I , which contains a strict subgroup \mathcal{Z} called the group of undetected syndromes. An element of \mathcal{P} is called a letter. An element of \mathcal{P}^n where $n \in \mathbb{N}^*$ is called an error. The weight of an error $E \in \mathcal{P}^n$ is $\#\{i \in \llbracket 1, n \rrbracket, E_i \neq I\}$, where E_i is the i -th coordinate of E .

- An $\llbracket n, k \rrbracket$ encoder, where $n \geq k$, is an isomorphism from $\mathcal{P}^k \times \mathcal{P}^{n-k}$ to \mathcal{P}^n .
- Let $E \in \mathcal{P}^n$, $E \neq I^n$, and C an $\llbracket n, k \rrbracket$ encoder. For $1 \leq i \leq n$, let $C^{-1}(E)_i$ be the i -th coordinate of $C^{-1}(E)$. Then E is undetected for C if:

$$\forall i \in \llbracket k+1, n \rrbracket, C^{-1}(E)_i \in \mathcal{Z}$$

E is harmless for C if it is undetected for C and:

$$\forall i \in \llbracket 1, k \rrbracket, C^{-1}(E)_i = I$$

E is harmful for C if it is undetected for C and:

$$\exists i \in \llbracket 1, k \rrbracket : C^{-1}(E)_i \neq I$$

- The distance d_c of C is the minimum weight of a harmful error for C
- The degenerate distance d_q of C is the minimum weight of an undetected error for C

Let us also introduce a few writing conventions. As presented, an error E is a sequence of elements of \mathcal{P} . The i -th element of this sequence is written E_i and we write $E = E_1.E_2.\dots.E_N$ where N is the size of E . E will also often be seen as a concatenation of errors of smaller size, each playing a particular role with respect to the encoding protocol. We have already seen the standard encoding protocol, in which the first k positions of the input carry the information, the last $n - k$ positions are an ancilla, and the n positions of the output carry the encoded information. Restrictions of an error to these respective positions are called information, stabilizer, and physical errors, and written L , S and P , such that in a standard encoding protocol, the error affecting the input can be written $E = (L, S)$, and the output can be written $C(E) = P$. As will be presented with the convolutional encoder, some positions play the role of a memory, in which case the restriction of an error to these positions is written M . The weight of the information, ancilla, physical and memory parts of E are written respectively $|E|_L$, $|E|_S$, $|E|_P$ and $|E|_M$. Moreover, the number of elements of E in the ancilla part which belong to $\mathcal{P} \setminus \mathcal{Z}$ is called the weight of the detected syndromes and written $|E|_X$.

3 The inner encoder: definitions and properties

3.1 Convolutional encoders, truncated convolutional decoders

Let us now define convolutional encoders and truncated convolutional decoders. Instead of considering simply the inverse transformation of a convolutional encoder, we will remove the stabilizer part of the output, and this is why we use the word truncated. Likewise, a similarity will be drawn between these two transformations, from which we can generalize simply what happens in a recursive encoder to what happens in a totally recursive decoder. A convolutional encoder is built by using a seed $\llbracket n, k, m \rrbracket$ encoder C . The seed transformation related to a truncated convolutional decoder is a truncated decoder:

Definition 3.1. An $\llbracket n, k, m \rrbracket$ encoder of memory size m , information size k , and stabilizer size $n - k$, is an isomorphism from $\mathcal{P}^m \times \mathcal{P}^k \times \mathcal{P}^{n-k}$ to $\mathcal{P}^n \times \mathcal{P}^m$. The truncated decoder of an $\llbracket n, k, m \rrbracket$ encoder C , is the application \bar{C} from $\mathcal{P}^m \times \mathcal{P}^n$ to $\mathcal{P}^k \times \mathcal{P}^m$ obtained by truncating the output of C^{-1} to its information and memory parts, more precisely, for all $(M', P) \in \mathcal{P}^m \times \mathcal{P}^n$, if:

$$C^{-1}(P, M') = (M, L, S)$$

then:

$$\bar{C}(M', P) = (L, M)$$

One very important thing to notice here is that an encoder and a truncated decoder are both particular cases of a morphism from $\mathcal{P}^m \times \mathcal{P}^k \times \mathcal{P}^s$ to $\mathcal{P}^n \times \mathcal{P}^m$. For an encoder, we have $n \geq k$ and $s = n - k$, whereas for a truncated decoder, we have $s = 0$ and the roles of k and n are switched. Let us thus say that a morphism from $\mathcal{P}^m \times \mathcal{P}^k \times \mathcal{P}^s$ to $\mathcal{P}^n \times \mathcal{P}^m$ is an $\llbracket n, k, s, m \rrbracket$ morphism. In this case, an $\llbracket n, k, m \rrbracket$ encoder is an $\llbracket n, k, n - k, m \rrbracket$ morphism, and the truncated decoder associated to an $\llbracket n, k, m \rrbracket$ encoder is an $\llbracket k, n, 0, m \rrbracket$ morphism.

A convolutional encoder C_N of size N and parameters (m, k, n) is an isomorphism from $\mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^{n-k})^{\otimes N}$ to $(\mathcal{P}^n)^{\otimes N} \times \mathcal{P}^m$. An input error is the concatenation of a memory error with N alternations of one information and one stabilizer error, and an output error is the concatenation of N physical errors ended by a memory error. The convolutional encoding is done by repeating N times a given $\llbracket n, k, m \rrbracket$ encoder C , which acts each time on one memory error, one information error and one stabilizer error, and produces one physical error and one memory error. Thus at the i -th step, the intermediate error is the concatenation of $i - 1$ physical errors, one memory error, and $N - i$ couples of information and stabilizer errors. Thus formally, we define a convolutional encoder by introducing N intermediate transformations acting on a total of $N + 1$ intermediate groups. A truncated convolutional decoder \bar{C}_N is a morphism from $\mathcal{P}^m \times (\mathcal{P}^n)^{\otimes N}$ to $(\mathcal{P}^k)^{\otimes N} \times \mathcal{P}^m$, obtained by applying successively a truncated decoder \bar{C} . Its input is a memory error concatenated with N physical errors, and its output is the concatenation of N information errors ended by a memory error. Let us define a convolutional encoder and a truncated convolutional decoder, after defining, for each of them, the $N + 1$ intermediate groups and the N intermediate transformations.

Definition 3.2. Let C be an $\llbracket n, k, m \rrbracket$ encoder and let $N \in \mathbb{N}^*$. For $1 \leq i \leq N + 1$, let:

$$\mathcal{P}_{i,N} = (\mathcal{P}^n)^{\otimes i-1} \times \mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^{n-k})^{\otimes N-i+1}$$

and let:

$$\mathcal{P}_{-i,N} = (\mathcal{P}^k)^{\otimes N-i+1} \times \mathcal{P}^m \times (\mathcal{P}^n)^{\otimes i-1}$$

For $1 \leq i \leq N$, let C_N^i be the isomorphism from $\mathcal{P}_{i,N}$ to $\mathcal{P}_{i+1,N}$ defined by:

$$C_N^i(P_1, \dots, P_{i-1}, M_{i-1}, L_i, S_i, \dots, L_N, S_N) = (P_1, \dots, P_i, M_i, L_{i+1}, S_{i+1}, \dots, L_N, S_N)$$

where

$$(P_i, M_i) = C(M_{i-1}, L_i, S_i)$$

Let also \bar{C}_N^i be the morphism from $\mathcal{P}_{-i-1,N}$ to $\mathcal{P}_{-i,N}$ defined by:

$$\bar{C}_N^i(L_N, \dots, L_{i+1}, M_i, P_i, \dots, P_1) = (L_N, \dots, L_i, M_{i-1}, P_{i-1}, \dots, P_1)$$

where

$$(L_i, M_{i-1}) = \bar{C}(M_i, P_i)$$

The convolutional encoder C_N is the isomorphism from $\mathcal{P}_{1,N}$ to $\mathcal{P}_{N+1,N}$ defined by:

$$C_N = C_N^N \circ C_N^{N-1} \circ \dots \circ C_N^1$$

The truncated convolutional decoder \bar{C}_N is the morphism from $\mathcal{P}_{-N-1,N}$ to $\mathcal{P}_{-1,N}$ defined by:

$$\bar{C}_N = \bar{C}_N^1 \circ \dots \circ \bar{C}_N^{N-1} \circ \bar{C}_N^N$$

Notice that if one runs the truncated convolutional decoder \bar{C}_N on the output of the convolutional encoder C_N , all the intermediate errors M_i during the truncated convolutional decoding stage correspond to the M_i of the convolutional encoding stage, and the information errors obtained at the end of the truncated decoding are equal to the initial information errors before the encoding. The memory errors M_i for $1 \leq i \leq N$ will play an important role later; let us call M_i the i th intermediate memory error. Also, let us call i th intermediate sequence the sequence $(P_1, \dots, P_i, M_i, L_{i+1}, S_{i+1}, \dots, L_N, S_N)$ obtained at the step i of the convolutional encoding. A convolutional encoder and a truncated convolutional decoder are both a case of a *convolutional morphism* defined as follows.

Definition 3.3. Let \mathcal{C} be an $\llbracket n, k, s, m \rrbracket$ morphism and let $N \in \mathbb{N}^*$. For $1 \leq i \leq N+1$, let:

$$\mathcal{P}_{i,N} = (\mathcal{P}^n)^{\otimes i-1} \times \mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^s)^{\otimes N-i+1}$$

For $1 \leq i \leq N$, let \mathcal{C}_N^i be the isomorphism from $\mathcal{P}_{i,N}$ to $\mathcal{P}_{i+1,N}$ defined by:

$$\mathcal{C}_N^i(P_1, \dots, P_{i-1}, M_{i-1}, L_i, S_i, \dots, L_N, S_N) = (P_1, \dots, P_i, M_i, L_{i+1}, S_{i+1}, \dots, L_N, S_N)$$

where

$$(P_i, M_i) = \mathcal{C}(M_{i-1}, L_i, S_i)$$

The convolutional morphism \mathcal{C}_N is the isomorphism from $\mathcal{P}_{1,N}$ to $\mathcal{P}_{N+1,N}$ defined by:

$$\mathcal{C}_N = \mathcal{C}_N^N \circ \mathcal{C}_N^{N-1} \circ \dots \circ \mathcal{C}_N^1$$

It is useful to separate the physical part and the memory part of an error output by a convolutional morphism:

Definition 3.4. Let $N \in \mathbb{N}^*$. π_N and μ_N are the maps from $\mathcal{P}_{1,N}$ to respectively $(\mathcal{P}^n)^{\otimes N}$ and \mathcal{P}^m , such that $\mathcal{C}_N = (\pi_N, \mu_N)$.

We also define a convolutional encoder C_∞ and truncated decoder \bar{C}_∞ for inputs of infinite size. They are both a particular case of the following convolutional morphism.

Definition 3.5. The convolutional morphism \mathcal{C}_∞ is the morphism from $\mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^s)^{\otimes \mathbb{N}}$ to $(\mathcal{P}^n)^{\otimes \mathbb{N}}$, such that for any infinite input sequence $E = (M, L_1, S_1, L_2, S_2, \dots)$, and for all $i \in \mathbb{N}^*$, the sequence of the first i errors of $C_\infty(E)$ is equal to $\pi_i(M, L_1, S_1, \dots, L_i, S_i)$.

The following lemma is a direct consequence of the convolutional construction:

Lemma 3.6. For all $1 \leq i \leq N$, and for all input sequences $(M, L_1, S_1, \dots, L_N, S_N)$:

$$C_N(M, L_1, S_1, \dots, L_N, S_N) = \pi_i(M, L_1, S_1, \dots, L_i, S_i). \\ C_{N-i}(\mu(M, L_1, S_1, \dots, L_i, S_i), L_{i+1}, S_{i+1}, \dots, L_N, S_N)$$

3.2 Recursive, systematic encoders, and totally recursive decoders

We define here the notion of a *recursive* encoder and a *systematic* encoder, which are an extension of the already existing notions known for classical convolutional encoders. We also present the new notion of a *total recursive* decoder. The word *total* is used to emphasize the fact that for a decoder to be *totally recursive*, no condition is required about the stabilizer weight of the output sequence of the decoder.

Definition 3.7. An $\llbracket n, k, s, m \rrbracket$ morphism \mathcal{C} is recursive if:

$$\forall E \in \mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^s)^{\otimes \mathbb{N}}, |E|_M = 0, |E|_L = 1 \text{ and } |E|_X = 0 \Rightarrow |\mathcal{C}_\infty(E)| = \infty$$

An $\llbracket n, k, m \rrbracket$ encoder C is recursive, or a truncated decoder \bar{C} is totally recursive, if the $\llbracket n, k, s, m \rrbracket$ morphism they correspond to is recursive. In other words, an $\llbracket n, k, m \rrbracket$ encoder C is recursive if:

$$\forall E \in \mathcal{P}^m \times (\mathcal{P}^k \times \mathcal{P}^{n-k})^{\otimes \mathbb{N}}, |E|_M = 0, |E|_L = 1 \text{ and } |E|_X = 0 \Rightarrow |C_\infty(E)| = \infty$$

An $\llbracket n, k, m \rrbracket$ truncated decoder \bar{C} is totally recursive if:

$$\forall E \in \mathcal{P}^m \times (\mathcal{P}^n)^{\otimes \mathbb{N}}, |E|_M = 0 \text{ and } |E|_P = 1 \Rightarrow |\bar{C}_\infty(E)| = \infty$$

An $\llbracket n, k, m \rrbracket$ encoder C is systematic if:

$$\forall N \in \mathbb{N}, \forall E \in \mathcal{P}_{1,N}, |C_N(E)| \geq |E|_L$$

An equivalent characterization of recursiveness will come after defining the sets \mathbb{M}_0 and \mathbb{M}_1 .

3.3 Memory errors, speed, and characterisation of recursive encoders

In this subsection, let \mathcal{C} be an $\llbracket n, k, s, m \rrbracket$ morphism. Memory errors have two kinds of behaviour. Consider an infinite input sequence E composed by concatenating a memory error M and infinitely many couples of one information error and one stabilizer error, where all the information errors are equal to I , and all the stabilizer errors belong to \mathcal{Z}^s . Then depending on M , the corresponding output will either have an infinite weight independently of the sequence of stabilizer errors, or there will exist a stabilizer sequence for which the output weight is finite. This is what we intend to describe in the definition of these two sets:

Definition 3.8.

$$\begin{aligned}\mathbb{M}_0 &= \{M \in \mathcal{P}^m / \exists (S_1, S_2, \dots) \in \mathcal{Z}^{s \otimes \mathbb{N}} / |\mathcal{C}_\infty(M, I, S_1, I, S_2, \dots)| < \infty\} \\ \mathbb{M}_1 &= \{M \in \mathcal{P}^m / \forall (S_1, S_2, \dots) \in \mathcal{Z}^{s \otimes \mathbb{N}}, |\mathcal{C}_\infty(M, I, S_1, I, S_2, \dots)| = \infty\}\end{aligned}$$

Let us also define a set useful for the characterization of a recursive morphism, the set of memory errors \mathbb{I} , accessible by starting with the I memory error and applying \mathcal{C} a finite number of times on an input of information weight 0 and detected syndrome weight 0:

Definition 3.9. $\mathbb{I} = \{ \mu_N(I, I, S_1, \dots, I, S_N), N \in \mathbb{N}, (S_1, \dots, S_N) \in (\mathcal{Z}^s)^N \}$

The goal of the next two lemmas is to prove that if a memory error belongs to \mathbb{M}_1 , then the weight of a finite output $\mathcal{C}_N(I, I, S_1, \dots, I, S_N)$ will be proportional to its length N . The constant of proportionality is in the form $1/\eta$ where the integer η depends on \mathcal{C} .

Lemma 3.10. *There exists a positive integer η such that:*

$$\forall M \in \mathbb{M}_1, \forall (S_1, \dots, S_\eta) \in (\mathcal{Z}^s)^{\otimes \eta}, |\pi_\eta(M, I, S_1, \dots, I, S_\eta)| \geq 1$$

The smallest such η is called the speed of the morphism \mathcal{C} .

Proof. We prove the lemma by contradiction by supposing that:

$$\forall \eta \in \mathbb{N}^*, \exists M_\eta \in \mathbb{M}_1, \exists (S_1^\eta, \dots, S_\eta^\eta) \in (\mathcal{Z}^s)^{\otimes \eta} / |\pi_\eta(M_\eta, I, S_1^\eta, \dots, I, S_\eta^\eta)| = 0$$

Since the sequence $(M_\eta)_{\eta \in \mathbb{N}}$ has its values in the finite set \mathbb{M}_1 , there exists $M \in \mathbb{M}_1$ such that $|\{\eta \in \mathbb{N}^* / M_\eta = M\}| = \infty$.

Now let us define a sequence of stabilizer errors $(S_i)_{i \in \mathbb{N}^*} \in (\mathcal{Z}^s)^{\mathbb{N}}$ such that:

$$\forall i \in \mathbb{N}^*, |\{\eta \geq i : (S_1^\eta, \dots, S_i^\eta) = (S_1, \dots, S_i)\}| = \infty$$

- The sequence $(S_1^\eta)_{\eta \in \mathbb{N}}$ has its values in the finite set \mathcal{Z}^s . Thus, define S_1 as an element of \mathcal{Z}^s such that:

$$|\{\eta \geq 1 : S_1^\eta = S_1\}| = \infty$$

- Suppose that for a given $i \in \mathbb{N}^*$, we have defined i stabilizer errors S_1, \dots, S_i , such that for all $j \leq i$:

$$|\{\eta \geq j : (S_1^\eta, \dots, S_j^\eta) = (S_1, \dots, S_j)\}| = \infty$$

Let $\mathbb{N}_i = \{\eta \geq i + 1 : (S_1^\eta, \dots, S_i^\eta) = (S_1, \dots, S_i)\}$, which by hypothesis is infinite. Since the sequence $(S_{i+1}^\eta)_{\eta \in \mathbb{N}_i}$ has its values in the finite set \mathcal{Z}^s , we can define S_{i+1} as an element of \mathcal{Z}^s such that:

$$|\{\eta \in \mathbb{N}_i : S_{i+1}^\eta = S_{i+1}\}| = \infty$$

which implies:

$$|\{\eta \geq i + 1 : (S_1^\eta, \dots, S_{i+1}^\eta) = (S_1, \dots, S_{i+1})\}| = \infty$$

Now let us use this sequence $(S_i)_{i \in \mathbb{N}^*} \in (\mathcal{Z}^s)^\mathbb{N}$ to produce a contradiction. Let $i \in \mathbb{N}^*$, and let $\eta \geq i$ such that $(S_1^\eta, \dots, S_i^\eta) = (S_1, \dots, S_i)$. Then:

$$|\pi_i(M, I, S_1, \dots, S_i)| = |\pi_i(M, I, S_1^\eta, \dots, S_i^\eta)| = 0$$

Since this is true for all $i \in \mathbb{N}^*$, $|\mathcal{C}_\infty(M, I, S_1, I, S_2, \dots)| = 0$, which contradicts the fact that $M \in \mathbb{M}_1$. \square

Lemma 3.11. *For all $N > 0$, and for all $(S_1, \dots, S_N) \in (\mathcal{Z}^s)^{\otimes \mathbb{N}}$ the function:*

$$\begin{aligned} \mathcal{P}^m &\rightarrow \mathcal{P}^m \\ M &\mapsto \mu_N(M, I, S_1, \dots, I, S_N) \end{aligned}$$

stabilizes the set \mathbb{M}_1 .

Proof. Let $N > 0$ and $(S_1, \dots, S_N) \in (\mathcal{Z}^s)^{\otimes N}$. Let $M \in \mathbb{M}_1$, and let $(P, M') = C_N(M, I, S_1, \dots, I, S_N)$. For all $(S_{N+1}, S_{N+2}, \dots) \in (\mathcal{Z}^s)^{\otimes \mathbb{N}}$:

$$\mathcal{C}_\infty(M, I, S_1, I, S_2, \dots) = (P, \mathcal{C}_\infty(M', I, S_{N+1}, I, S_{N+2}, \dots))$$

And since $M \in \mathbb{M}_1$, $|\mathcal{C}_\infty(M', I, S_{N+1}, I, S_{N+2}, \dots)| = \infty$.

Thus $M' \in \mathbb{M}_1$. \square

As a consequence:

Lemma 3.12. *For all $N > 0$, $M \in \mathbb{M}_1$, and $(S_1, \dots, S_N) \in (\mathcal{Z})^{s \otimes N}$:*

$$|\pi_N(M, I, S_1, \dots, I, S_N)| \geq \lfloor N/\eta \rfloor$$

Proof. Let $M \in \mathbb{M}_1$. We prove the lemma by recursion on $p = \lfloor N/\eta \rfloor$.

For $p = 0$, the property is clearly true.

Now suppose the property is true for a given $p - 1$, where $p \geq 1$.

Let $N > 0$ such that $\lfloor N/\eta \rfloor = p$, and let $(S_1, \dots, S_N) \in (\mathcal{Z})^{s \otimes N}$. The following relation holds:

$$\pi_N(M, I, S_1, \dots, I, S_N) = \pi_\eta(M, I, S_1, \dots, I, S_\eta) \cdot \pi_{N-\eta}(\mu_\eta(M, I, S_1, \dots, I, S_\eta), I, S_{\eta+1}, \dots, I, S_N)$$

Since $M \in \mathbb{M}_1$, $\mu_\eta(M, I, S_1, \dots, I, S_\eta) \in \mathbb{M}_1$, which implies by the recursion hypothesis:

$$|\pi_{N-\eta}(\mu_\eta(M, I, S_1, \dots, I, S_\eta), I, S_{\eta+1}, \dots, I, S_N)| \geq p - 1$$

And since $|\pi_\eta(M, I, S_1, \dots, I, S_\eta)| \geq 1$, the property is also true for p . \square

Another consequence is the following characterization of recursive morphisms:

Lemma 3.13. *\mathcal{C} is recursive if and only if for all $(M, L, S) \in \mathbb{I} * \mathcal{P}^k * \mathcal{Z}^s$ with $|L| = 1$, $\mu(M, L, S) \in \mathbb{M}_1$.*

Proof. Let E be an infinite input sequence of memory weight 0, information weight 1, and such that all its stabilizer letters are in \mathcal{Z}^s . Let K be the position of the only information error L of weight 1. E is in the form:

$$E = (I, I, S_1, \dots, I, S_{i-1}, L, S_i, I, S_{i+1}, I, S_{i+2}, \dots)$$

Let $(P, M) = \mathcal{C}_{i-1}(I, I, S_1, \dots, I, S_{i-1})$. Then:

$$\begin{aligned} \mathcal{C}_\infty(E) &= P. \mathcal{C}_\infty(M, L, S_i, I, S_{i+1}, I, S_{i+2}, \dots) \\ &= P. \pi(M, L, S_i). \mathcal{C}_\infty(\mu(M, L, S_i), I, S_{i+1}I, S_{i+2}, \dots) \end{aligned}$$

Thus, $|\mathcal{C}_\infty(E)| = \infty$ for all E which verify the assumptions above, if and only if $\mu(M, L, S_i) \in \mathbb{M}_1$ for all $i \geq 1$, L of weight 1, S_i in \mathcal{Z}^s , and M in the form $\mu_{i-1}(I, I, S_1, \dots, I, S_{i-1})$ where $(S_1, \dots, S_{i-1}) \in (\mathcal{Z}^s)^{i-1}$, or in other words for all $M \in \mathbb{I}$. □

4 Weight distribution of the inner encoder: upper bounds in the recursive and the totally recursive cases

In this section, we will establish two upper bounds to the weight distribution of a convolutional encoder. What is meant precisely by the weight distribution is the number of possible (memory and information) subsequences (M, L_1, \dots, L_N) of an input sequence $E = (M, L_1, S_1, \dots, L_N, S_N)$, for a given weight w of the input subsequence and a given weight d of the output sequence. The first upper bound comes provided that the seed encoder \mathcal{C} is recursive, and the second one comes provided that the seed truncated decoder $\bar{\mathcal{C}}$ is totally recursive. We will establish an upper bound for a recursive morphism, and derive the two desired results as a corollary.

4.1 Trace and detours

Before going to the point of the upper bounds, we first need to exhibit a characteristic behaviour of recursive morphisms; mainly, we will first show that the *trace* of an input sequence of a convolutional morphism is a concatenation of *detours*. By introducing the concept of *trace*, this phenomenon of *detours* is a generalisation of the phenomenon described by [1] in the case of a classical convolutional encoder based on a recursive seed encoder. Consider a recursive $\llbracket n, k, s, m \rrbracket$ morphism \mathcal{C} . Let N be a positive integer, and consider $E = (M, L_1, S_1, \dots, L_N, S_N) = E_1 \dots E_N$ an input sequence to the convolutional morphism \mathcal{C}_N , where $\mathcal{N} = m + N(k + s)$. Thus we have $M \in \mathcal{P}^m$, and for all $i \in \llbracket 1, n \rrbracket$, $L_i \in \mathcal{P}^k$ and $S_i \in \mathcal{Z}^s$. Let $(P_1, \dots, P_N, M') = \mathcal{C}_N(E)$ be the output sequence.

Let $w_M = |E|_M = |M|$ and $w_L = |E|_L = |(L_1, \dots, L_N)|$. By looking at the information part (L_1, \dots, L_N) as a sequence of kN letters, label by $1 \leq p_1 < \dots < p_{w_L} \leq kN$ the positions

of the w_L non identity letters, and for each $i \in \llbracket 1; w_L \rrbracket$, let $N_i = \lceil p_i/k \rceil$ so that the i th non identity letter is part of the error L_{N_i} .

Define the i th truncature of E as the sequence:

$$E_{\setminus i} = (M, L_1, S_1, \dots, L_{N_{i-1}}, S_{N_{i-1}}, L'_{N_i}, S_{N_i})$$

where $(L_1, \dots, L_{N_{i-1}}, L'_{N_i})$ is obtained by replacing all the letters in (L_1, \dots, L_{N_i}) after the position p_i by an I . Let us also set $p_0 = 0$, $N_0 = 0$ and $E_{\setminus 0} = M$. Define M_i as the memory error of the output when running \mathcal{C}_{N_i} on $E_{\setminus i}$:

$$M_i = \mu_{N_i}(E_{\setminus i})$$

Set also $M_0 = M$. This enables to define the trace of E :

Definition 4.1. *The trace of E is the sequence (b_0, \dots, b_{w_L}) of elements of $\{0, 1\}$ such that for all $i \in \llbracket 0; w_L \rrbracket$, $M_i \in \mathbb{M}_{b_i}$.*

The following result holds as a direct consequence of this definition:

Lemma 4.2. *$b_i = 1$ if and only if for all sequences $(\tilde{S}_K)_{K \in \mathbb{N}^*}$ of stabilizer errors in \mathcal{Z}^s :*

$$|C_\infty(E_{\setminus i} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots))| = \infty$$

Proof. $b_i = 1$ if and only if $M_i \in \mathbb{M}_1$, thus if and only if for all sequences $(\tilde{S}_K)_{K \in \mathbb{N}^*}$ of stabilizer errors in \mathcal{Z}^s :

$$|C_\infty(M_i, I, \tilde{S}_1, I, \tilde{S}_2, \dots)| = \infty$$

The relations:

$$C_\infty(E_{\setminus i} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots)) = \pi_{N_i}(E_{\setminus i}) \cdot C_\infty(M_i, I, \tilde{S}_1, I, \tilde{S}_2, \dots)$$

if $i \geq 1$, and if $i = 0$:

$$C_\infty(E_{\setminus 0} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots)) = C_\infty(M_0, I, \tilde{S}_1, I, \tilde{S}_2, \dots)$$

show that this happens if and only if $|C_\infty(E_{\setminus i} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots))| = \infty$. □

This implies the following property about the trace of E :

Lemma 4.3. *For all i in $\llbracket 0, w_L - 1 \rrbracket$, if $b_i = 0$ then $b_{i+1} = 1$.*

Proof. Let us suppose that $b_i = 0$. Then there exists a sequence $(S'_K)_{K \in \mathbb{N}^*}$ of stabilizer errors in \mathcal{Z}^s such that:

$$|C_\infty(E_{\setminus i} \cdot (I, S'_1, I, S'_2, \dots))| < \infty$$

Let $(\tilde{S}_K)_{K \in \mathbb{N}^*}$ be a sequence of stabilizer errors in \mathcal{Z}^s . Then the two sequences

$$E_{\setminus i} \cdot (I, S'_1, I, S'_2, \dots)$$

and

$$E_{\setminus i+1} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots)$$

do not differ in their memory part, differ in their information parts by only the p_{i+1} th letter of (L_1, \dots, L_N) , and differ in their stabilizer parts by a sequence of errors in \mathcal{Z}^s . Thus one can write:

$$E_{\setminus i+1} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots) = E_{\setminus i} \cdot (I, S'_1, I, S'_2, \dots) * \Delta E$$

where the sequence ΔE verifies $|\Delta E|_M = 0$, $|\Delta E|_L = 1$ and $|\Delta E|_X = 0$. Since \mathcal{C} is recursive:

$$|\mathcal{C}_\infty(\Delta E)| = \infty$$

And since \mathcal{C}_∞ is a morphism:

$$\mathcal{C}_\infty(E_{\setminus i+1} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots)) = \mathcal{C}_\infty(E_{\setminus i} \cdot (I, S'_1, I, S'_2, \dots)) * \mathcal{C}_\infty(\Delta E)$$

We deduce that:

$$|\mathcal{C}_\infty(E_{\setminus i+1} \cdot (I, \tilde{S}_1, I, \tilde{S}_2, \dots))| = \infty$$

And since this is true for all sequences $(\tilde{S}_K)_{K \in \mathbb{N}^*}$ of stabilizer errors in \mathcal{Z}^s , $b_{i+1} = 1$. \square

Definition 4.4. A *detour* is a finite and non empty sequence of 1, potentially completed with a 0 in which case it is called a *terminating detour*.

If $w_L \geq 1$, according to the previous lemma, either b_0 or b_1 is equal to 1. Starting from that first 1, the rest of the trace of E is a concatenation of detours, which are all terminating except maybe the last detour:

Lemma 4.5. If $w_L \geq 1$, there exist $c \geq 1$ and c integers $v_1 < \dots < v_c \leq w_L$ such that:

- $v_1 = 0$ or 1
- for all i in $\llbracket 0; c-1 \rrbracket$, $(b_{v_i}, \dots, b_{v_{i+1}-1})$ is a terminating detour
- $(b_{v_c}, \dots, b_{w_L})$ is a detour

As a convention, let $v_{c+1} = w_L + 1$, so that $(b_{v_i}, \dots, b_{v_{i+1}-1})$ is also the expression of a detour if $i = c$.

The number c of detours is less or equal to $\lfloor \frac{w_L}{2} + 1 \rfloor$. Indeed for each $i \in \llbracket 1; c-1 \rrbracket$, the i th detour contains at least a 1 and a 0, whereas the last detour contains at least a 1; this implies that $w_L + 1 \geq 2(c-1) + 1$, in other words $c \leq w_L/2 + 1$.

4.2 The upper bounds

We are still with an $\llbracket n, k, s, m \rrbracket$ recursive morphism \mathcal{C} , and we note η its speed. Let $a_N(w, \leq d)$ (for $d \in \mathbb{R}$) and $a_N(w, d)$ (for $d \in \mathbb{N}$) be the numbers of sequences $(M, L_1, \dots, L_N) \in \mathcal{P}^m * (\mathcal{P}^k)^N$ of weight w , which are part of an undetected input sequence:

$$E = (M, L_1, S_1, \dots, L_N, S_N) \in \mathcal{P}^m * (\mathcal{P}^k * \mathcal{Z}^s)^N$$

such that, respectively, $|C_N(E)| \leq d$ and $|C_N(E)| = d$. The above characterisation of detours enables to prove the result (the proof is given in appendix):

Theorem 4.6.

$$a_N(w, d) \leq a_N(w, \leq d) \leq 2^m 2^w (|\mathcal{P}| - 1)^w \binom{kN + 1}{\lfloor \frac{w}{2} \rfloor + 1} \binom{\eta k(w + d) + 1}{\lceil \frac{w}{2} \rceil}$$

Notice that $a_N(d, w)$ in the case where \mathcal{C} is the truncated decoder \bar{C} corresponds to the number of possible sequences (P_1, \dots, P_N, M') of weight d such that the weight of $(M, L_1, \dots, L_N) = \bar{C}_N(P_1, \dots, P_N, M')$ has weight w . This corresponds exactly to the value of $a_N(w, d)$ in the case where \mathcal{C} is the encoder C . This is what will enable us to obtain the two upper bounds on $a_N(w, d)$ concerning the encoder C . By using the following bound on binomials, where $v \leq u$:

$$\binom{u}{v} \leq \left(\frac{u \cdot e}{v}\right)^v$$

the two upper bounds can be written in the form of this corollary:

Corollary 4.7. *Let C be an encoder, and let $a_N(w, \leq d)$ and $a_N(w, d)$ be the number of possible sequences (M, L_1, \dots, L_N) of weight w , which are part of an input sequence $E = (M, L_1, S_1, \dots, L_N, S_N)$ where all the S_i belong to \mathcal{Z}^s , and such that $C_N(E)$ has weight respectively less or equal to d , and equal to d . Then:*

Bound 1I: *If C is recursive:*

$$a_N(w, \leq d) \leq O(1)^w \frac{N^{\frac{w}{2}} (w + d)^{\frac{w}{2}}}{w^w}$$

Bound 2I: *If \bar{C} is totally recursive:*

$$a_N(w, d) \leq O(1)^d \frac{N^{\frac{d}{2}} (w + d)^{\frac{d}{2}}}{d^d}$$

5 The outer encoder

The outer encoder consists in an $\llbracket n, k \rrbracket$ encoder C repeated blockwisely N times (C and N are not the same as the inner encoder).

Definition 5.1. *Let C be a $\llbracket n, k \rrbracket$ encoder and let $N \in \mathbb{N}^*$. The blockwise encoder $C^{\otimes N}$ is an $\llbracket Nn, Nk \rrbracket$ encoder defined by:*

$$C^{\otimes N}(L_1, \dots, L_N, S_1, \dots, S_N) = C(L_1, S_1) \cdot \dots \cdot C(L_N, S_N)$$

where for all $i \leq N$, $L_i \in \mathcal{P}^k$ and $S_i \in \mathcal{P}^{n-k}$.

Lemma 5.2. Bound 1E: *Let C be an encoder of distance d_c and degenerate distance $d_q \geq 2$. For all $d > 0$ and $N > 0$, let $a^{\otimes N}(d)$ be the number of input sequences $E \in \mathcal{P}^{Nk} \times \mathcal{Z}^{N(n-k)}$ such that $|E|_L > 0$ and $|C^{\otimes N}(E)| = d$. Then:*

$$\begin{aligned} a^{\otimes N}(d) &\leq O(1)^d \left(\frac{N}{d}\right)^{\frac{d-d_c}{d_q}+1} && \text{if } d \geq d_c \\ &= 0 && \text{if } d < d_c \end{aligned}$$

Proof. Of course if $d > N$ then the bound is true since $a^{\otimes N}(d) = 0$, so we will consider that $d \leq N$. Let $E = (L_1, \dots, L_N, S_1, \dots, S_N)$ where $|E|_L > 0$, $(S_1, \dots, S_N) \in (\mathcal{Z}^{n-k})^N$, and such that $|C^{\otimes N}(E)| = d$. Let $j > 0$ be the number of couples (L_i, S_i) different from (I, I) . In other words all these j couples are undetected errors for C , thus we have $|C(L_i, S_i)| \geq d_q$. Also,

since $|E|_L > 0$, at least one of these couples is such that $L_i \neq I$. This couple is a harmful error for C which implies $|C(L_i, S_i)| \geq d_c$. This first shows that necessarily $d \geq d_c$, and:

$$d \geq (j-1)d_q + d_c$$

and thus:

$$j \leq \lfloor \frac{d-d_c}{d_q} \rfloor + 1$$

For each possible value for j , there are $\binom{N}{j}$ ways to chose the positions of the non zero sequences (L_i, S_i) . Each of these sequences can take less than $|\mathcal{P}|^n$ values whereas the remaining sequences are all fixed to (I, I) . This leads to the bound:

$$a^{\otimes N}(d) \leq \sum_{j=1}^{\lfloor \frac{d-d_c}{d_q} \rfloor + 1} |\mathcal{P}|^{nj} \binom{N}{j}$$

Since $\lfloor \frac{d-d_c}{d_q} \rfloor + 1 \leq \frac{d}{2} \leq \frac{N}{2}$, each term of the sum can be majored by the term where $j = \lfloor \frac{d-d_c}{d_q} \rfloor + 1$. Thus:

$$a^{\otimes N}(d) \leq (\lfloor \frac{d-d_c}{d_q} \rfloor + 1) |\mathcal{P}|^{n(\lfloor \frac{d-d_c}{d_q} \rfloor + 1)} \binom{N}{\lfloor \frac{d-d_c}{d_q} \rfloor + 1}$$

Finally the lemma is proved using the bound on binomials, where $v \leq u$:

$$\binom{u}{v} \leq \left(\frac{u.e}{v} \right)^v$$

□

Lemma 5.3. Bound 2E: Suppose that $|\mathcal{P}| > 2$. Let C be an encoder of degenerate distance $d_q \geq 2$. There exists a constant $c \in]0, 1[$ such that for all d and N :

$$a^{\otimes N}(d) \leq c^d (|\mathcal{P}| - 1)^d \binom{Nn}{d}$$

Proof. Let $X \in \mathcal{P} \setminus \mathcal{Z}$ (a non empty set since $|\mathcal{P}| > 2$), and let $Y = X * Z^{-1}$. For all $i \in \llbracket 0, n \rrbracket$, let:

$$\mathcal{C}_i = \{E \in C(\mathcal{P}^k * \mathcal{Z}^{n-k}), |E| = i\}$$

And let:

$$c_i = \frac{|\mathcal{C}_i|}{(|\mathcal{P}| - 1)^i \binom{n}{i}}$$

In particular, $c_0 = 1$. Let us show that for $i > 0$, $c_i < 1$, or in other words, that $C(\mathcal{P}^k * \mathcal{Z}^{n-k})$ does not contain all the sequences of weight i . Of course, $c_1 = 0$ since $d_q \geq 2$. Let us consider the case $i > 1$. If $C(\mathcal{P}^k * \mathcal{Z}^{n-k})$ contains the two sequences of weight i :

$$\begin{aligned} E_1 &= X^i . I^{n-i}, \text{ and} \\ E_2 &= Z . X^{i-1} . I^{n-i} \end{aligned}$$

then since it is a subgroup of \mathcal{P}^n , it contains the sequence of weight 1:

$$E_1 * E_2^{-1} = Y.I^{n-1}$$

This contradicts the fact that $d_q \geq 2$. Thus, E_1 and E_2 are not both in \mathcal{C}_i , and $c_i < 1$. Let $c = \max\{c_i^{1/i}, 1 \leq i \leq n\}$, so that $c \in]0, 1[$ and for all $i \in \llbracket 0, n \rrbracket$, $c_i \leq c^i$. Now consider an input sequence $E \in \mathcal{P}^{Nk} \times \mathcal{Z}^{Nn}$, where $|E|_L > 0$ and $|C^{\otimes N}(E)| = d$. The output $C^{\otimes N}(E)$ is the concatenation of N sequences in $C(\mathcal{P}^k * \mathcal{Z}^{n-k})$ such that the sum of their weights is equal to d . Thus:

$$a^{\otimes N}(d) \leq \sum_{\substack{(d_1, \dots, d_N) \\ \sum d_i = d \\ 0 \leq d_i \leq n}} \prod_{i=1}^N |\mathcal{C}_{d_i}| \leq \sum_{\substack{(d_1, \dots, d_N) \\ \sum d_i = d \\ 0 \leq d_i \leq n}} \prod_{i=1}^N c^{d_i} (|\mathcal{P}| - 1)^{d_i} \binom{n}{d_i}$$

The product of the terms c^{d_i} is always equal to c^d . The product of the remaining terms, summed over (d_1, \dots, d_N) , counts exactly the number of sequences of size Nn and weight d . This proves the lemma. \square

6 Upper bounds on the distance of a turbo-encoder

6.1 Formal definition of a turbo-encoder

A turbo-encoder T_N of size N is a concatenation of three operations. It requires a $\llbracket n_{out}, k_{out} \rrbracket$ encoder C_{out} , an interleaver Π , and an $\llbracket n_{in}, k_{in}, m_{in} \rrbracket$ encoder C_{in} . An interleaver is the following operation:

Definition 6.1. *An interleaver Π of size N is an automorphism of \mathcal{P}^N composed of two operations:*

- *First, a permutation π of the N positions of the letters of the sequence*
- *Second, a sequence (π_1, \dots, π_N) of N automorphisms of \mathcal{P} , applied at each one of the letters*

Thus an interleaver Π transforms a sequence $E = E_1. \dots .E_N$ into:

$$\pi_1(E_{\pi(1)}) . \dots . \pi_N(E_{\pi(N)})$$

The set of interleavers of size N is noted \mathbb{P}_N .

Definition 6.2. *Consider:*

- C_{out} an $\llbracket n_{out}, k_{out} \rrbracket$ encoder and C_{in} an $\llbracket n, k, m \rrbracket$ encoder
- $N = N_{out} \in \mathbb{N}^*$ and $N_{in} \in \mathbb{N}^*$ such that $N_{out}n_{out} = N_{in}k_{in} + m_{in}$
- Π an interleaver of size $N_{out}n_{out}$.

The turbo-encoder T_N based on C_{out} , Π , and C_{in} , is an $\llbracket N_{in}n_{in} + m_{in}, N_{out}k_{out} \rrbracket$ encoder which does the following transformation. Consider a sequence $E \in \mathcal{P}^{N_{out}k_{out}} \times \mathcal{P}^{N_{in}(n_{in}-k_{in})}$ written in the form:

$$E = (L_1, \dots, L_{N_{out}}, S_1, \dots, S_{N_{out}}, S'_1, \dots, S'_{N_{in}})$$

Then $T_N(E)$ is obtained by these three steps:

- First, apply $C_{out}^{\otimes N}$ at the first $2N$ errors of E :

$$E' = C_{out}^{\otimes N}(L_1, \dots, L_N, S_1, \dots, S_N)$$

- Then, apply Π at E' :

$$E'_{perm} = \Pi(E')$$

Since $N_{out}n_{out} = N_{in}k_{in} + m_{in}$, E'_{perm} can be written in the form:

$$E'_{perm} = (M', L'_1, \dots, L'_{N_{in}})$$

where M' is of size m_{in} and all the other errors L_i are of size k_{in} . Note that the first m_{in} letters of E'_{perm} are specialized into memory letters for the next step.

- Finally, insert $(S'_1, \dots, S'_{N_{in}})$ into E'_{perm} by putting each S'_i after L'_i , and apply $C_{inN_{in}}$:

$$T_N(E) = C_{inN_{in}}(M', L'_1, S'_1, \dots, L'_{N_{in}}, S'_{N_{in}})$$

Note that even the memory part of this last output counts as part of the physical output with respect to the global protocol.

A random turbo-encoder based on C_{out} and C_{in} is a turbo-encoder based on C_{out} , an interleaver Π chosen randomly with a uniform distribution over $\mathbb{P}_{Nn_{out}}$, and C_{in} . N is called the length of the turbo-encoder, and an integer N is said to be eligible for C_{out} and C_{in} if there exists an integer N_{in} such that $Nn_{out} = N_{in}k_{in} + m_{in}$.

If we go back to describing the real protocol lying behind a turbo-encoder, a state of information of size $N_{out}k_{out}$ is encoded into a state of size $N_{in}n_{in} + m_{in}$, by first encoding it into a state of size $N_{out}n_{out} = N_{in}k_{in} + m_{in}$, then interleaving the positions of the obtained state, then encoding it again into a state of size $N_{in}n_{in} + m_{in}$. At each of the first and the last steps of the encoding, ancillary positions are added before the encoding is done. This is what corresponds, in the formal protocol, to the insertion of the stabilizer errors S_i and S'_i .

From now on, for a given turbo-encoder T_N and a given input sequence E of the turbo-encoder, we will systematically write E' and E'_{perm} to refer to the intermediate sequences obtained after applying $C_{out}^{\otimes N}$ and after applying Π .

6.2 Sketch of the counting argument

We will sew the results obtained on the weight distributions for the inner encoder $a_{N_{in}}(w, d)$, $a_{N_{in}}(w, \leq d)$, and the outer encoder $a^{\otimes N}(d)$, $a^{\otimes N}(\leq d)$, to obtain the desired upper bounds on the distance of the turbo-encoder. For this purpose, the tools will be the two following lemmas. Let C_{out} be a $\llbracket n_{out}, k_{out} \rrbracket$ encoder and C_{in} be an $\llbracket n_{in}, k_{in}, m_{in} \rrbracket$ encoder. Let T_N be a random turbo-encoder based on C_{out} and C_{in} where N is an eligible integer for C_{out} and C_{in} . Consider the probabilities related to the following events:

- $p_N(d)$ and $p_N(\leq d)$: there exists a harmful input sequence E such that, respectively, $|T_N(E)| = d$ and $|T_N(E)| \leq d$. Note that $p_N(\leq d)$ is the probability that $d_c(T_N) \leq d$.
- $p_N(w, d)$ and $p_N(w, \leq d)$: there exists a harmful input sequence E such that $|E'| = w$ and, respectively, $|T_N(E)| = d$ and $|T_N(E)| \leq d$.

Since the interleaver Π is chosen at random, these probabilities are only functions of C_{out} , C_{in} and N . These probabilities are defined for all integers w and d , and $p_N(\leq d)$ and $p_N(w, \leq d)$ are also defined when d is real.

Lemma 6.3.

$$\begin{aligned} \forall (w, d) \in \mathbb{N} \times \mathbb{N}, p_N(w, d) &\leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, d)}{(|\mathcal{P}| - 1)^w \binom{Nn_{out}}{w}} \\ \forall (w, d) \in \mathbb{N} \times \mathbb{R}, p_N(w, \leq d) &\leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, \leq d)}{(|\mathcal{P}| - 1)^w \binom{Nn_{out}}{w}} \end{aligned}$$

Proof. Let us prove the first inequality, the second one can be obtained with a similar reasoning. $p_N(w, d)$ is the probability that there exists an input sequence E for the turbo-encoder T_N such that $|E|_L > 0$, $|E'| = w$ and $|T_N(E)| = d$. In order to obtain this, a necessary condition is that the first $2N$ errors of E constitute one of the $a^{\otimes N}(w)$ harmful input sequences of $C_{out}^{\otimes N}$ such that the output (by $C_{out}^{\otimes N}$) has weight w . Take such a sequence of $2N$ errors. E' is now uniquely defined. Since E'_{perm} is the image of E' under the action of a random interleaver Π , it is uniformly distributed over the set of sequences of size Nn_{out} and weight w . Having $|T_N(E)| = d$ implies that E'_{perm} is one of the $a_{N_{in}}(w, d)$ sequences of weight w which are part of an undetected input sequence of $C_{inN_{in}}$ such that the output by $C_{inN_{in}}$ has weight d . This has probability:

$$\frac{a_{N_{in}}(w, d)}{(|\mathcal{P}| - 1)^w \binom{Nn_{out}}{w}}$$

The inequality of the lemma is proved using the union bound, by summing this probability over all the $a^{\otimes N}(w)$ possible values of the first $2N$ errors. \square

The second lemma is straightforward by using the union bound:

Lemma 6.4. *For any real number D , and for all $x \in]0, n_{in}[$:*

$$p_N(\leq D) \leq \sum_{w=0}^{\lfloor D \rfloor} p_N(w, \leq d) + \sum_{d=0}^{\lfloor D \rfloor} \sum_{w=\lfloor D \rfloor + 1}^{\lfloor xN \rfloor - 1} p_N(w, d) + \sum_{d=0}^{\lfloor D \rfloor} \sum_{w=\lfloor xN \rfloor}^{Nn_{out}} p_N(w, d)$$

We call these three terms first, second and third partial sum.

Proof.

$$\begin{aligned}
p_N(\leq D) &\leq \sum_{w=0}^{Nn_{out}} p_N(w, \leq D) \\
&\leq \sum_{w=0}^{\lfloor D \rfloor} p_N(w, \leq D) + \sum_{w=\lfloor D \rfloor+1}^{Nn_{out}} p_N(w, \leq D) \\
&\leq \sum_{w=0}^{\lfloor D \rfloor} p_N(w, \leq D) + \sum_{d=0}^{\lfloor D \rfloor} \sum_{w=\lfloor D \rfloor+1}^{Nn_{out}} p_N(w, d) \\
&\leq \sum_{w=0}^{\lfloor D \rfloor} p_N(w, \leq D) + \sum_{d=0}^{\lfloor D \rfloor} \sum_{w=\lfloor D \rfloor+1}^{\lfloor xN \rfloor-1} p_N(w, d) + \sum_{d=0}^{\lfloor D \rfloor} \sum_{w=\lfloor xN \rfloor}^{Nn_{out}} p_N(w, d)
\end{aligned}$$

□

6.3 The polynomial bounds

We will start by proving case 1 of Theorem 1.1, then Theorem 1.2 since these results are close to each other, then we will prove case 2 of Theorem 1.1. The first two results are obtained using the three partial sums when $D = N^\alpha$. With the three following lemmas proved in Appendix B.1, we show successively that under the corresponding conditions, each of these three partial sums tends to 0 as $N \rightarrow \infty$. We write d_c and d_q respectively for the distance and the degenerate distance of C_{out} .

Lemma 6.5. First partial sum, poly case *If $d_q \geq 2$ and C_{in} is recursive, for all $\alpha < \frac{d_q-2}{d_q}$:*

$$\lim_{N \rightarrow \infty} \sum_{w=0}^{\lfloor N^\alpha \rfloor} p_N(w, \leq N^\alpha) = 0$$

Lemma 6.6. Second partial sum, poly case *If $d_q > 2$ and \bar{C}_{in} is totally recursive, then for all $\alpha < \frac{d_q-2}{d_q}$, there exists $x > 0$ such that :*

$$\lim_{N \rightarrow \infty} \sum_{d=0}^{\lfloor N^\alpha \rfloor} \sum_{w=\lfloor N^\alpha \rfloor+1}^{\lfloor xN \rfloor-1} p_N(w, d) = 0$$

Lemma 6.7. Third partial sum, poly case *If $|\mathcal{P}| > 2$, $d_q \geq 2$ and \bar{C}_{in} is totally recursive, then for all $x > 0$:*

$$\lim_{N \rightarrow \infty} \sum_{d=0}^{\lfloor N^\alpha \rfloor} \sum_{w=\lfloor xN \rfloor}^{Nn_{out}} p_N(w, d) = 0$$

The proof of Theorem 1.1 case 1 is now the following.

Proof. The hypothesis of the theorem are that $|\mathcal{P}| > 2$, $d_q > 2$, C_{in} is recursive and \bar{C}_{in} is totally recursive. The conditions are met so that the results from the lemmas above apply.

For all $\alpha < \frac{d_q-2}{d_q}$, let x such that the second partial sum goes to 0. Let us write:

$$p_N(\leq N^\alpha) \leq \sum_{w=0}^{\lfloor N^\alpha \rfloor} p_N(w, \leq d) + \sum_{d=0}^{\lfloor N^\alpha \rfloor} \sum_{w=\lfloor N^\alpha \rfloor+1}^{\lfloor xN \rfloor-1} p_N(w, d) + \sum_{d=0}^{\lfloor N^\alpha \rfloor} \sum_{w=\lfloor xN \rfloor}^{Nn_{out}} p_N(w, d)$$

Since the three partial sums go to 0, this proves that $p_N(\leq N^\alpha)$ goes to 0 as $N \rightarrow \infty$. \square

The proof of Theorem 1.2 is the following:

Proof. The hypothesis of the theorem are that $d_q > 2$, and C_{in} is recursive and systematic. Being systematic implies that if $w > d$, $p_N(w, d) = 0$. Thus the second and the third partial sums are equal to 0. The conditions of Lemma 6.5 are met and thus again, $p_N(\leq N^\alpha)$ goes to 0 as $N \rightarrow \infty$. \square

6.4 The sublogarithmic bound

Now let us focus on Theorem 1.1 case 2. For this purpose we consider the three partial sums when $D = \log \log N$. The following proposition justifies the sublogarithmic expression of the bound. This property is needed in order to upper bound the first partial sum.

Proposition 6.8. : *Let us write, for $N > 1$, $\text{llog } N = \log N / \log(\log N)$. For all $t > 0$, if N is big enough(precisely, if $N \geq e^{e^t}$):*

$$(t \text{llog } N)^{t \text{llog } N} \leq N^t$$

Proof. If $N \geq e^{e^t}$, $\text{llog}(N)$ is positive and its logarithm is well defined, and the logarithm of the left hand side of the inequality is equal to:

$$\begin{aligned} (t \text{llog } N)(\log t + \log(\text{llog } N)) &= t \frac{\log N}{\log(\log N)} (\log t + \log(\log N) - \log(\log(\log N))) \\ &= t \log N + t \frac{\log N}{\log(\log N)} \log \left(\frac{t}{\log \log N} \right) \\ &\leq t \log N \end{aligned}$$

\square

Actually one can prove that the solution of the equation $x^x = N^t$ is equivalent to $t \log N / \log(\log N)$, and thus the bound proposed is the best asymptotic distance which can be proved with the arguments presented in this paper. The three following lemmas proved in Appendix B.2 show under the corresponding conditions that each of the three partial bounds, when $D = \log \log N$, tend to 0 as $N \rightarrow \infty$.

Lemma 6.9. First partial sum, sub-log case *If $d_c > d_q = 2$ and C_{in} is recursive, for all $\alpha < d_c - 2$:*

$$\lim_{N \rightarrow \infty} \sum_{w=0}^{\lfloor \alpha \text{llog } N \rfloor} p_N(w, \leq \alpha \text{llog } N) = 0$$

Lemma 6.10. Second partial sum, sub-log case *If $d_c > d_q = 2$ and \bar{C}_{in} is totally recursive, for all $\alpha < d_c - 2$, there exists x such that:*

$$\lim_{N \rightarrow \infty} \sum_{d=0}^{\lfloor \alpha \log N \rfloor} \sum_{w=\lfloor \alpha \log N \rfloor + 1}^{\lfloor xN \rfloor - 1} p_N(w, d) = 0$$

Lemma 6.11. Third partial sum, sub-log case *If $|\mathcal{P}| > 2$, $d_q \geq 2$ and \bar{C}_{in} is totally recursive, then for all $x > 0$:*

$$\lim_{N \rightarrow \infty} \sum_{d=0}^{\lfloor \alpha \log N \rfloor} \sum_{w=\lfloor xN \rfloor}^{N n_{out}} p_N(w, d) = 0$$

The proof of Theorem 1.1 case 2 is now the following.

Proof. The hypothesis of the theorem are that $|\mathcal{P}| > 2$, $d_c > d_q = 2$, C_{in} is recursive and \bar{C}_{in} is totally recursive. The conditions are met so that the results from the lemmas above apply. For all $\alpha < d_c - 2$, let x such that the second partial sum goes to 0. Let us write:

$$p_N(\alpha \log N) \leq \sum_{w=0}^{\lfloor \alpha \log N \rfloor} p_N(w, \leq \alpha \log N) + \sum_{d=0}^{\lfloor \alpha \log N \rfloor} \sum_{w=\lfloor \alpha \log N \rfloor + 1}^{\lfloor xN \rfloor - 1} p_N(w, d) + \sum_{d=0}^{\lfloor \alpha \log N \rfloor} \sum_{w=\lfloor xN \rfloor}^{N n_{out}} p_N(w, d)$$

Since the three partial sums go to 0, this proves that $p_N(\alpha \log N)$ goes to 0 as $N \rightarrow \infty$. \square

Appendix A: Bounds for the inner convolutional encoder

We give here the proof of Theorem 4.6. Consider a sequence (M, L_1, \dots, L_N) of weight w , which is part of $E = (M, L_1, S_1, \dots, L_N, S_N)$ such that $|\mathcal{C}_N(E)| \leq d$, and let $(P_1, \dots, P_N, M') = \mathcal{C}_N(E)$. To begin with, suppose that the information part is of given weight $|(L_1, \dots, L_N)| = w_L$, and suppose that the trace of E is made of c detours. We remind the following notations. $1 \leq p_1 < \dots < p_{w_L} \leq kN$ are the positions of the w_L non identity letters of (L_1, \dots, L_N) , and for each $i \in \llbracket 1; w_L \rrbracket$, $N_i = \lceil p_i/k \rceil$ is the index such that the i th non identity letter is part of the error L_{N_i} . The i th truncature of E is the sequence:

$$E_{\setminus i} = (M, L_1, S_1, \dots, L_{N_{i-1}}, S_{N_{i-1}}, L'_{N_i}, S_{N_i})$$

where $(L_1, \dots, L_{N_{i-1}}, L'_{N_i})$ is obtained by replacing all the letters in (L_1, \dots, L_{N_i}) after the position p_i by an I . Also, $p_0 = 0$, $N_0 = 0$ and $E_{\setminus 0} = M$. Moreover:

$$M_i = \mu_{N_i}(E_{\setminus i})$$

and for $i = 0$, $M_0 = M$. The trace of E is the sequence (b_0, \dots, b_{w_L}) such that for $i \in \llbracket 1; w_L \rrbracket$, $M_i \in \mathbb{M}_{b_i}$. For $i \in \llbracket 1; c \rrbracket$, v_i is the starting point of the i th detour, and $v_{c+1} = w_L + 1$.

We will start by confining the space where lie the positions of the non identity letters of (L_1, \dots, L_N) , by cutting them into packets corresponding to each detour. For each $i \in \llbracket 1; c \rrbracket$, let $\delta p^{(i)} = p_{v_{i+1}-1} - p_{v_i}$. The sum of the sizes of these intervals is upper bounded by:

Lemma 6.12. *If $w_L \geq 1$:*

$$\sum_{i=1}^c \delta p^{(i)} \leq \min(kN, \eta k(w_L + d))$$

where $\delta p^{(i)} = p_{v_{i+1}-1} - p_{v_i}$.

The main idea is to prove that the difference of weight between the outputs at the steps N_{v_i} and $N_{v_{i+1}}$ of the convolutional operation is proportional to $\delta p^{(i)}$. The main argument is that for j from v_i to $v_{i+1} - 1$, the memory errors M_j all belong to \mathbb{M}_1 , which by the recursiveness of \mathcal{C} yields an output weight proportional to the input size during that time. We first apply this idea between the steps N_j and $N_{j+1} - 1$ of the encoding for a given $j \in \llbracket v_0; w_L \rrbracket$:

Sublemma 6.13. *Let $j \in \llbracket v_0; w_L \rrbracket$.*

$$(|P_{N_j+1}, \dots, P_{N_{j+1}-1}|) \geq \frac{N_{j+1} - N_j}{\eta} - 1$$

Proof. If $N_{j+1} = N_j$ or $N_j + 1$, this is true since the sequence $(P_{N_j+1}, \dots, P_{N_{j+1}-1})$ is empty, and its weight is non negative.

Now suppose that $N_{j+1} > N_j + 1$, and let us first consider the case $j > 0$. Notice that $L'_{N_j} = L_{N_j}$, because the non identity information letter in position p_{j+1} belongs to the error $L_{N_{j+1}}$, which comes strictly after the error L_{N_j} . We can write the following concatenation:

$$E_{\setminus j+1} = E_{\setminus j} \cdot (I, S_{N_j+1}, \dots, I, S_{N_{j+1}-1}, L'_{N_{j+1}}, S_{N_{j+1}})$$

The physical output $(P_1, \dots, P_{N_{j+1}-1})$ is produced by applying the encoder at the first $2N_{j+1}-1$ errors of $E_{\setminus j+1}$. By concatenation, this output can be written:

$$\begin{aligned} (P_1, \dots, P_{N_{j+1}-1}) &= \pi_{N_{j+1}-1}(E_{\setminus j} \cdot (I, S_{N_j+1}, \dots, I, S_{N_{j+1}-1})) \\ &= \pi_{N_j}(E_{\setminus j}) \cdot \pi_{N_{j+1}-N_j-1}(M_j, I, S_{N_j+1}, \dots, I, S_{N_{j+1}-1}) \\ &= (P_1, \dots, P_{N_j}) \cdot \pi_{N_{j+1}-N_j-1}(M_j, I, S_{N_j+1}, \dots, I, S_{N_{j+1}-1}) \end{aligned}$$

This shows that:

$$(P_{N_j+1}, \dots, P_{N_{j+1}-1}) = \pi_{N_{j+1}-N_j-1}(M_j, I, S_{N_j+1}, \dots, I, S_{N_{j+1}-1})$$

In the case where $j = 0$, this equality also holds and has the form:

$$(P_1, \dots, P_{N_1-1}) = \pi_{N_1-1}(M_0, I, S_1, \dots, I, S_{N_1-1})$$

Since $M_j \in \mathbb{M}_1$ and \mathcal{C} is recursive:

$$(|P_{N_j+1}, \dots, P_{N_{j+1}-1}|) \geq \lfloor \frac{N_{j+1} - N_j - 1}{\eta} \rfloor$$

And this implies the desired inequality. □

Now by simply summing over j we obtain the proof of the lemma.

Proof. Let $i \in \llbracket 1; c \rrbracket$. The sum of the inequality in the previous lemma over all the values of $j \in \llbracket v_i; v_{i+1} - 2 \rrbracket$ gives the result:

$$|(P_{N_{v_i}}, \dots, P_{N_{v_{i+1}-1}})| \geq \frac{N_{v_{i+1}-1} - N_{v_i}}{\eta} - (v_{i+1} - v_i - 1)$$

Using $N_{v_i} \leq p_{v_i}/k + 1$ and $N_{v_{i+1}-1} \geq p_{v_{i+1}-1}/k$ we get:

$$|(P_{N_{v_i}}, \dots, P_{N_{v_{i+1}-1}})| \geq \frac{p_{v_{i+1}-1} - p_{v_i}}{\eta k} - \frac{1}{\eta} - (v_{i+1} - v_i - 1)$$

By summing again this inequality over $i \in \llbracket 1; c \rrbracket$ we get that:

$$d \geq \sum_{i=1}^c \frac{\delta p^{(i)}}{\eta k} - w_L$$

This implies the first bound:

$$\sum_{i=1}^c \delta p^{(i)} \leq \eta k(w_L + d)$$

The second bound is obvious:

$$\sum_{i=1}^c \delta p^{(i)} = \sum_{i=1}^c (p_{v_{i+1}-1} - p_{v_i}) \leq p_{w_L} - p_{v_c} + \sum_{i=1}^{c-1} (p_{v_{i+1}} - p_{v_i}) \leq kN$$

□

This upper bound is useful in that it says that all the non identity letters of (L_1, \dots, L_N) starting from the letter in position p_{w_1} (ie, from the first non identity letter since $w_1 = 0$ or 1) are confined in a bounded region of space. Now we can prove the following bound:

Lemma 6.14. *Suppose $w_L \geq 1$, and suppose that c is fixed. The number of possible values for the sequence (L_1, \dots, L_N) is upper bounded by:*

$$(|\mathcal{P}| - 1)^{w_L} \binom{w_L}{c-1} \binom{kN}{c} \binom{\min(kN, \eta k(w_L + d)) + 1}{w_L - c + 1}$$

Proof. There are two possible values for the starting point w_1 of the first detour, either 0 or 1. In both cases, the number of possible starting points w_i , $2 \leq i \leq c$ of the $c - 1$ remaining detours is upper bounded by $\binom{w_L}{c-1}$, since the $c - 1$ remaining values are in the interval $\llbracket 1, v_c \rrbracket$. For each choice of these starting points, the number of possible values for the positions $(p_{v_i})_{1 \leq i \leq c}$, is at most $\binom{kN}{c}$.

Now, consider that the sequences $(v_i)_{1 \leq i \leq c}$ and $(p_{v_i})_{1 \leq i \leq c}$ are fixed. The number of positions of the non identity letters in (L_1, \dots, L_N) which are still unfixed is equal to $w_L - c$ if $v_1 = 0$ and $w_L - c + 1$ if $v_1 = 1$. This sequence of remaining positions can be written:

$$(p_j)_{v_i < j < v_{i+1}, 1 \leq i \leq c}$$

where as defined previously, $v_{c+1} - 1 = w_L$. In order to upper bound the number of such possible remaining sequences, we will rely on the fact that $\sum_{i=1}^c \delta p^{(i)}$, which we know is

upper bounded, is intuitively the space in which the remaining positions are confined. Since this sequence is strictly growing, it is equivalent to upper bound the number of possible corresponding sets. Such a corresponding set can be written as:

$$\bigcup_{i=1}^c \mathcal{I}_i$$

where:

$$\mathcal{I}_i = \{p_j, v_i < j < v_{i+1}\}.$$

In the following, we exhibit a reversible transformation of this set into a set of equal number of elements and included in the interval $\llbracket 1, \sum_{i=1}^c \delta p^{(i)} \rrbracket$. Define $s_1 = 0$ and for $i \in \llbracket 2, c+1 \rrbracket$, let $s_i = \sum_{i'=1}^{i-1} \delta p^{(i')}$. Consider the transformation which transforms the set of remaining positions into :

$$\bigcup_{i=1}^c \bar{\mathcal{I}}_i$$

where:

$$\bar{\mathcal{I}}_i = \{\bar{p}_j = p_j - p_{v_i} + s_i, v_i < j < v_{i+1}\}.$$

This function relies on the knowledge of v_i and p_{v_i} . Let us show that this transformation is injective by showing explicitly how to recover the antecedent of a given image set.

First, let us show that in the image set, any two elements \bar{p}_j and $\bar{p}_{j'}$ respect the order of their indexes, ie they verify $\bar{p}_j < \bar{p}_{j'}$ if and only if $j < j'$.

For each $i \in \llbracket 1, c \rrbracket$, and for each couple $(\bar{p}_j, \bar{p}_{j'}) \in \bar{\mathcal{I}}_i^2$ where $j \leq j'$:

$$\begin{array}{ll} \bar{p}_j \geq 1 + s_i & \text{because } p_j - p_{v_i} \geq 1 \\ \bar{p}_{j'} \leq s_{i+1} & \text{because } p_{j'} - p_{v_i} \leq \delta p^{(i)} \\ \bar{p}_j \leq \bar{p}_{j'} & \text{because } p_j \leq p_{j'} \end{array}$$

If we compare two elements in a same set $\bar{\mathcal{I}}_i$, the third inequality proves that they respect the order of their indexes, whereas if $\bar{p}_j \in \bar{\mathcal{I}}_i$ and $\bar{p}_{j'} \in \bar{\mathcal{I}}_{i'}$ with $i < i'$, we can use the first two inequalities to get:

$$\bar{p}_j \leq s_{i+1} \leq 1 + s_{i'} \leq 1 + \bar{p}_{j'}$$

This implies that, by looking at the position of an element respectively to the others in the image set, we know necessarily the value of its corresponding index j . Next, the $c+1$ values of s_i can be recovered recursively as follows. We start with $s_1 = 0$. Then, for each $i \in \llbracket 1, c \rrbracket$, either $\bar{\mathcal{I}}_i$ is empty, in which case $v_{i+1} = v_i + 1$ and $s_{i+1} = s_i$, or it is not, in which case s_{i+1} is equal to the last element of $\bar{\mathcal{I}}_i$: $\bar{p}_{v_{i+1}-1} = \delta p^{(i)} + s_i = s_{i+1}$. Now that all the values of p_j are known, and since the values of p_{v_i} are also known, all the values of p_j can be recovered. Finally, the fact that all the values \bar{p}_j are confined in the set $\llbracket 1, \sum_{i=1}^c \delta p^{(i)} \rrbracket$ comes from the first two inequalities.

This proves respectively that, when $v_1 = 0$ and when $v_1 = 1$, the number of possible sequences of remaining positions $(p_j)_{v_i < j < v_{i+1}, 1 \leq i \leq c}$ is upper bounded by the number of

subsets of $\llbracket 1, \min(kN, \eta k(w_L + d)) \rrbracket$ of respective number of elements $w_L - c$ and $w_L - c + 1$. This gives two binomials, the sum of which is:

$$\binom{\min(kN, \eta k(w_L + d)) + 1}{w_L - c + 1}$$

Finally, the proof is completed by noticing that each of the w_L non identity letters takes at most $|\mathcal{P}| - 1$ values. \square

Now let us release the constraint on the number of detours c :

Lemma 6.15. *Suppose that $w_L \geq 0$. The number of possible values for the sequence (L_1, \dots, L_N) is upper bounded by:*

$$2^{w_L} (|\mathcal{P}| - 1)^{w_L} \binom{kN + 1}{\lfloor \frac{w_L}{2} \rfloor + 1} \binom{\min(kN, \eta k(w_L + d)) + 1}{\lceil \frac{w_L}{2} \rceil}$$

Proof. The sequence of N identity errors is the only one such that $w_L = 0$. Thus if $w_L = 0$, the bound is true. Now we suppose that $w_L \geq 1$. For a purpose of lisibility, let $\mathcal{B} = \min(kN, \eta k(w_L + d)) + 1$, and let $c_{max} = \lfloor \frac{w_L}{2} \rfloor + 1$ be the maximal value for the number c of detours. By the previous lemma, the number of possible values for the sequence (L_1, \dots, L_N) is upper bounded by:

$$(|\mathcal{P}| - 1)^{w_L} \sum_{c=1}^{c_{max}} \binom{w_L}{c-1} \max_{1 \leq c \leq c_{max}} \binom{kN}{c} \binom{\mathcal{B}}{w_L - c + 1}$$

Replacing kN by $kN + 1$ in the previous expression keeps the inequality true and makes the analysis easier. Let us maximize over c , and under the constraint $c \leq c_{max}$, the expression:

$$f(c) = \binom{kN + 1}{c} \binom{\mathcal{B}}{w_L - c + 1}$$

For all $c \leq c_{max} - 1$, the ratio $r(c)$ between two successive values is:

$$r(c) = \frac{f(c+1)}{f(c)} = \frac{kN + 1 - c}{\mathcal{B} - (w_L - c)} \frac{w_L - c + 1}{c + 1}$$

When $c \leq c_{max} - 1$, $c \leq w_L/2$, and thus $c \leq w_L - c$. Besides, $\mathcal{B} \leq kN + 1$. Thereby, $r(c) \geq 1$ for all $c \leq c_{max} - 1$, which proves that the maximum is reached at $c = c_{max}$. When c takes this value, $w_L - c + 1$ is equal to $\lceil \frac{w_L}{2} \rceil$; this comes from the fact that $\lfloor \frac{w_L}{2} \rfloor + \lceil \frac{w_L}{2} \rceil = w_L$. Thus:

$$\max_{1 \leq c \leq c_{max}} \binom{kN}{c} \binom{\mathcal{B}}{w_L - c + 1} = \binom{kN + 1}{\lfloor \frac{w_L}{2} \rfloor + 1} \binom{\min(kN, \eta k(w_L + d)) + 1}{\lceil \frac{w_L}{2} \rceil}$$

Moreover:

$$\sum_{c=1}^{c_{max}} \binom{w_L}{c-1} \leq 2^{w_L}$$

and this proves the lemma. \square

Finally comes the proof of Theorem 4.6:

Proof. The number of memory errors M of given weight w_M is:

$$(|\mathcal{P}| - 1)^{w_M} \binom{m}{w_M}$$

Thus by the previous lemma, the number of possible sequences (M, L_1, \dots, L_N) verifying the conditions of the theorem and such that $|M| = w_M$ is upper bounded by:

$$(|\mathcal{P}| - 1)^w \binom{m}{w_M} g(w_L)$$

where $w_L = w - w_M$ and where the function g is given by:

$$g(w_L) = 2^{w_L} \binom{kN + 1}{\lfloor \frac{w_L}{2} \rfloor + 1} \binom{\eta k(w + d) + 1}{\lceil \frac{w_L}{2} \rceil}$$

Again, we search for the maximum of g for all values of w_L under the constraint $w_L \leq w \leq kN$. If $w_L \leq w - 1$ then these two inequalities:

$$\lfloor \frac{w_L}{2} \rfloor + 1 \leq \frac{1}{2}(kN + 1)$$

and:

$$\lceil \frac{w_L}{2} \rceil \leq \frac{1}{2}(\eta k(w + d) + 1)$$

show that $g(w_L + 1) > g(w_L)$. Then, g reaches its maximum at $w_L = w$. The proof is completed by the majoration:

$$\sum_{w_M=0}^m \binom{m}{w_M} \leq 2^m$$

□

Appendix B: partial sums

B.1: Polynomial case

Proof of Lemma 6.5:

Proof. First partial sum, poly case Let $d = N^\alpha$. We use the following property:

$$p_N(w, \leq d) \leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, \leq d)}{(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w}}$$

with the following bound derived from the bound (1E), valid because $d_q \geq 2$:

$$\begin{aligned} a^{\otimes N}(w) &\leq O(1)^w \left(\frac{N}{w} \right)^{\frac{w}{d_q}} && \text{if } w \geq d_c \\ &= 0 && \text{if } w < d_c \end{aligned}$$

and the bound (1I), valid because C_{in} is recursive:

$$a_{N_{in}}(w, \leq d) \leq O(1)^w \frac{N_{in}^{\frac{w}{2}}(w + d)^{\frac{w}{2}}}{w^w}$$

Using the fact that $w \leq d$ and $N_{in} = O(N)$, this bound gives :

$$a_{N_{in}}(w, \leq d) \leq O(1)^w \frac{N^{\frac{w}{2}} d^{\frac{w}{2}}}{w^w}$$

We lower bound the denominator using the following lower bound on binomials:

$$\forall (u, v) \in \mathbb{N}^2 / v \leq u, \binom{u}{v} \geq \left(\frac{u}{v}\right)^v$$

which, together with the facts that $|\mathcal{P}| \geq 2$ and $n_{out} \geq 1$, implies :

$$(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w} \geq \left(\frac{N}{w}\right)^w$$

As a consequence we obtain, when $w \geq d_c$:

$$\begin{aligned} p_N(w, \leq d) &\leq O(1)^w \left(\frac{N}{w}\right)^{\frac{w}{d_q}} \frac{N^{\frac{w}{2}} d^{\frac{w}{2}}}{w^w} \left(\frac{N}{w}\right)^{-w} \\ &\leq O(1)^w N^{\frac{w}{d_q}} N^{\frac{w}{2}} d^{\frac{w}{2}} N^{-w} \\ &\leq \left[O(1) N^{\frac{1}{d_q} - \frac{1}{2}} d^{\frac{1}{2}}\right]^w \end{aligned}$$

and when $w < d_c$, $p_N(w, \leq d) = 0$. Let us simply replace d by its value N^α :

$$p_N(w, \leq N^\alpha) \leq \left[O(1) N^{\frac{1}{d_q} + \frac{\alpha}{2} - \frac{1}{2}}\right]^w$$

The exponent verifies:

$$\frac{1}{d_q} + \frac{\alpha}{2} - \frac{1}{2} = \frac{1}{2} \left(\frac{2 - d_q}{d_q} + \alpha \right) < 0$$

And since $p_N(w, \leq N^\alpha) = 0$ when $w < d_c$, this implies:

$$\begin{aligned} \sum_{w=0}^{\lfloor N^\alpha \rfloor} p_N(w, \leq N^\alpha) &\leq \sum_{w=d_c}^{\infty} \left(O(1) N^{\frac{1}{d_q} + \frac{\alpha}{2} - \frac{1}{2}} \right)^w \\ &\leq O(1) N^{(\frac{1}{d_q} + \frac{\alpha}{2} - \frac{1}{2})d_c} \end{aligned}$$

This proves that the first partial sum tends to 0 as $N \rightarrow \infty$. □

Proof of Lemma 6.6:

Proof. Second partial sum, poly case We use:

$$p_N(w, d) \leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, d)}{(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w}}$$

with the same bound as before derived from (1E), valid because $d_q \geq 2$:

$$\begin{aligned} a^{\otimes N}(w) &\leq O(1)^w \left(\frac{N}{w}\right)^{\frac{w}{d_q}} && \text{if } w \geq d_c \\ &= 0 && \text{if } w < d_c \end{aligned}$$

and with the bound (2I), valid because \bar{C}_{in} is totally recursive:

$$a_{N_{in}}(w, d) \leq O(1)^d \frac{N_{in}^{\frac{d}{2}}(w + d)^{\frac{d}{2}}}{d^d}$$

Since $N_{in} = O(N)$ and $w > d$, this last bound can be written in the following form:

$$a_{N_{in}}(w, d) \leq O(1)^w \frac{N^{\frac{d}{2}} w^{\frac{d}{2}}}{d^d}$$

Let us also lower bound the denominator as previously by:

$$\left(\frac{N}{w}\right)^w$$

As a consequence:

$$\begin{aligned} p_N(w, d) &\leq O(1)^w \left(\frac{N}{w}\right)^{\frac{w}{d_q}} \frac{N^{\frac{d}{2}} w^{\frac{d}{2}}}{d^d} \left(\frac{N}{w}\right)^{-w} \\ &\leq \left[O(1) \left(\frac{N}{w}\right)^{\frac{1}{d_q} - 1}\right]^w \frac{N^{\frac{d}{2}} w^{\frac{d}{2}}}{d^d} \end{aligned}$$

Let us upper bound $O(1)$ by a constant $a > 0$, and take the logarithm of the last inequality:

$$\log(p_N(w, d)) \leq f(w, d) = w(\log a + (1 - 1/d_q)(\log w - \log N)) + \frac{d}{2}(\log N + \log w - 2\log d)$$

Let us show that there exists x sufficiently small such that, for N sufficiently large, the maximum of $f(w, d)$ in the domain $(w, d) \in \llbracket \lfloor N^\alpha \rfloor, \lfloor xN \rfloor - 1 \rrbracket \times \llbracket 0, \lfloor N^\alpha \rfloor \rrbracket$ is reached when $w = d = \lfloor N^\alpha \rfloor$. First, consider the derivative of f with respect to w :

$$\log a + (1 - 1/d_q)(\log w - \log N + 1) + \frac{d}{2} \frac{1}{w}$$

Since $\frac{d}{2} \frac{1}{w} \leq \frac{1}{2}$ and $w \leq xN$, this is upper bounded by:

$$\log a + (1 - 1/d_q)(\log x + 1) + \frac{1}{2}$$

For a sufficiently small x , this upper bound is negative. Indeed, when $x \rightarrow 0$, the above expression is dominated by the term:

$$(1 - 1/d_q) \log x$$

which tends to $-\infty$. This shows that, provided x is small enough, $f(w, d)$ is maximum when $w = \lfloor N^\alpha \rfloor$. Now, suppose that $w = \lfloor N^\alpha \rfloor$, and consider the derivative of f with respect to d :

$$\frac{1}{2}(\log N + \log \lfloor N^\alpha \rfloor - 2\log d - 2)$$

This is a decreasing function of d . Thus the minimum of this derivative is reached at $d = \lfloor N^\alpha \rfloor$ and is equal to:

$$\frac{1}{2}(\log N - \log \lfloor N^\alpha \rfloor - 2)$$

When $N \rightarrow \infty$ this minimum is equivalent to:

$$\frac{1-\alpha}{2} \log N$$

Since $\alpha < 1$, for a sufficiently large N this minimum is positive. Consequently $f(w, d)$ is maximum when $w = d = \lfloor N^\alpha \rfloor$. The maximum of f in the domain is equal to:

$$\begin{aligned} f(\lfloor N^\alpha \rfloor, \lfloor N^\alpha \rfloor) &= \lfloor N^\alpha \rfloor (\log a + (1 - 1/d_q)(\log \lfloor N^\alpha \rfloor - \log N)) + \frac{\lfloor N^\alpha \rfloor}{2} (\log N - \log \lfloor N^\alpha \rfloor) \\ &= \lfloor N^\alpha \rfloor (\log a + (1/2 - 1/d_q)(\log \lfloor N^\alpha \rfloor - \log N)) \end{aligned}$$

When $N \rightarrow \infty$ this maximum is equivalent to:

$$f(\lfloor N^\alpha \rfloor, \lfloor N^\alpha \rfloor) \sim -(1/2 - 1/d_q)(1 - \alpha)N^\alpha \log N$$

Now in the sum:

$$\sum_{d=0}^{\lfloor N^\alpha \rfloor} \sum_{w=\lfloor N^\alpha \rfloor+1}^{\lfloor xN \rfloor-1} p_N(w, d)$$

the number of terms is upper bounded by xN^2 . Thus the logarithm of the sum is upper bounded by:

$$\log x + 2 \log N + f(\lfloor N^\alpha \rfloor, \lfloor N^\alpha \rfloor) \sim -(1/2 - 1/d_q)(1 - \alpha)N^\alpha \log N$$

This equivalent tends to $-\infty$, which proves that the second partial sum tends to 0 if we chose a sufficiently small x . \square

Proof of Lemma 6.7:

Proof. Third partial sum, poly case Again we use:

$$p_N(w, d) \leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, d)}{(|\mathcal{P}| - 1)^w \binom{Nn_{out}}{w}}$$

According to the bound (2E), valid because $|\mathcal{P}| > 2$ and $d_q \geq 2$, there exists a constant $c \in]0, 1[$ such that:

$$a^{\otimes N}(w) \leq c^w (|\mathcal{P}| - 1)^w \binom{Nn_{out}}{w}$$

Let us also use the bound (2I), valid because C_{in}^τ is totally recursive:

$$a_{N_{in}}(w, d) \leq O(1)^d \frac{N_{in}^{\frac{d}{2}}(w + d)^{\frac{d}{2}}}{d^d}$$

Since $N_{in} = O(N)$, $d \leq N$ and $w \leq Nn_{out}$, this bound gives:

$$a_{N_{in}}(w, d) \leq O(1)^d N^d$$

With these two bounds we obtain:

$$p_N(w, d) \leq c^w O(1)^d N^d$$

Thus:

$$\begin{aligned}\log(p'_N(w, d)) &\leq w \log c + d(O(1) + \log N) \\ &\leq \lfloor xN \rfloor \log c + N^\alpha(O(1) + \log N)\end{aligned}$$

where the last line is obtained using $\log c < 0$, $w \geq \lfloor xN \rfloor$ and $d \leq N^\alpha$. Since the third partial sum contains less than $n_{out}N^2$ terms, its logarithm is upper bounded by:

$$\log n_{out} + 2 \log N + \log(p'_N(w, d)) \leq \log n_{out} + 2 \log N + \lfloor xN \rfloor \log c + N^\alpha(O(1) + \log N)$$

Since $\alpha < 1$, this upper bound is equivalent to:

$$xN \log c$$

and this proves that the sum tends to 0 when $N \rightarrow \infty$. □

B.2: Sub-logarithmic case

Proof of Lemma 6.9

Proof. First partial sum, sub-log case Let $d = \alpha \log N$. We use:

$$p_N(w, \leq d) \leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, \leq d)}{(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w}}$$

Let us use this result from the bound (1E), valid because $d_q \geq 2$:

$$a^{\otimes N}(w) \leq O(1)^w \left(\frac{N}{w} \right)^{\frac{w-dc}{2} + 1}$$

and the bound (1I), valid because C_{in} is recursive:

$$a_{N_{in}}(w, \leq d) \leq O(1)^w \frac{N_{in}^{\frac{w}{2}} (w + d)^{\frac{w}{2}}}{w^w}$$

Since $w \leq d$ and $N_{in} = O(N)$ this bound gives:

$$a_{N_{in}}(w, \leq d) \leq O(1)^w \frac{N^{\frac{w}{2}} d^{\frac{w}{2}}}{w^w}$$

Let us also lower bound the denominator:

$$(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w} \geq \left(\frac{N}{w} \right)^w$$

By combining these three bounds we get:

$$\begin{aligned}p_N(w, \leq d) &\leq O(1)^w \left(\frac{N}{w} \right)^{\frac{w-dc}{2} + 1} \frac{N^{\frac{w}{2}} d^{\frac{w}{2}}}{w^w} \left(\frac{N}{w} \right)^{-w} \\ &\leq O(1)^w N^{\frac{w-dc}{2} + 1} N^{\frac{w}{2}} d^{\frac{w}{2}} N^{-w} \\ &\leq [O(1)d]^{\frac{w}{2}} N^{-\frac{dc}{2} + 1}\end{aligned}$$

The last bound is an increasing function of w . Applied at $w = \alpha \log N$, it is thus a bound for each term of the first partial sum:

$$\sum_{w=0}^{\lfloor \alpha \log N \rfloor} p_N(w, \leq \alpha \log N) \leq (\alpha \log N + 1) [O(1) \alpha \log N]^{\frac{\alpha \log N}{2}} N^{-\frac{d_c}{2} + 1}$$

Now, notice that:

$$O(1)^{\frac{\alpha \log N}{2}} = \exp \left(O(1) \frac{\alpha \log N}{2 \log(\log N)} \right) = N^{\frac{O(1)}{\log(\log N)}}$$

And using Proposition 6.8:

$$[\alpha \log N]^{\frac{\alpha \log N}{2}} \leq N^{\frac{\alpha}{2}}$$

Mutlplying these terms we get:

$$\sum_{w=0}^{\lfloor \alpha \log N \rfloor} p_N(w, \leq \alpha \log N) \leq (\alpha \log N + 1) N^{\frac{O(1)}{\log(\log N)}} N^{\frac{\alpha - d_c}{2} + 1}$$

Since $\frac{\alpha - d_c}{2} + 1 < 0$ the first partial sum tends to 0 when $N \rightarrow \infty$. □

Proof of Lemma 6.10

Proof. Second partial sum, sub-log case We use:

$$p_N(w, d) \leq \frac{a^{\otimes N}(w) a_{N_{in}}(w, d)}{(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w}}$$

Let us apply this result from the bound (1E), valid because $d_q \geq 2$:

$$a^{\otimes N}(w) \leq O(1)^w \left(\frac{N}{w} \right)^{\frac{w - d_c}{2} + 1}$$

and the bound (2I), valid because \bar{C}_{in} is totally recursive:

$$a_{N_{in}}(w, d) \leq O(1)^d \frac{N_{in}^{\frac{d}{2}} (w + d)^{\frac{d}{2}}}{d^d}$$

Using the facts that $N_{in} = O(N)$ and $w > d$, this bound gives:

$$a_{N_{in}}(w, d) \leq O(1)^w \frac{N^{\frac{w}{2}} w^{\frac{d}{2}}}{d^d}$$

We also lower bound the denominator:

$$(|\mathcal{P}| - 1)^w \binom{N n_{out}}{w} \geq \left(\frac{N}{w} \right)^w$$

Combining these bounds gives:

$$\begin{aligned} p_N(w, d) &\leq O(1)^w \left(\frac{N}{w} \right)^{\frac{w - d_c}{2} + 1} \frac{N^{\frac{w}{2}} w^{\frac{d}{2}}}{d^d} \left(\frac{N}{w} \right)^{-w} \\ &\leq N^{-\frac{d_c}{2} + 1} \left[O(1) \left(\frac{N}{w} \right)^{-\frac{1}{2}} \right]^w \frac{N^{\frac{d}{2}} w^{\frac{d}{2}}}{d^d} \end{aligned}$$

Let us upper bound $O(1)$ by a constant $a > 0$; then consider the logarithm $f(w, d)$ of the right hand expression apart from the first factor:

$$\begin{aligned} f(w, d) &= \log \left(\left[a \left(\frac{N}{w} \right)^{-\frac{1}{2}} \right]^w \frac{N^{\frac{d}{2}} w^{\frac{d}{2}}}{d^d} \right) \\ &= w(\log a + 1/2(\log w - \log N)) + \frac{d}{2}(\log N + \log w - 2 \log d) \end{aligned}$$

Let us show that there exists x sufficiently small such that, for N sufficiently large, the maximum of $f(w, d)$ in the domain $(w, d) \in \llbracket \lfloor \alpha \log N \rfloor; \lfloor xN \rfloor - 1 \rrbracket \times \llbracket 0; \lfloor \alpha \log N \rfloor \rrbracket$ is reached when $w = d = \lfloor \alpha \log N \rfloor$. The derivative of f with respect to w is equal to:

$$\log a + 1/2(\log w - \log N + 1) + \frac{d}{2w} \leq \log a + 1/2(\log x + 1) + \frac{1}{2}$$

where the inequality comes from the facts that $w \leq xN$ and $d \leq w$. For a sufficiently small x , the right hand side (and thus the partial derivative) is negative. This shows that the maximum of $f(w, d)$ is reached if $w = \lfloor \alpha \log N \rfloor$. Moreover, there exists $r \in]0; 1[$ such that the derivative of f with respect to w is upper bounded by $\log r$. By integrating this inequality from $\lfloor \alpha \log N \rfloor$ to w we get the following inequality:

$$f(w, d) \leq f(\lfloor \alpha \log N \rfloor, d) + \log r (w - \lfloor \alpha \log N \rfloor) \quad (1)$$

Now suppose that $w = \lfloor \alpha \log N \rfloor$, and consider the derivative of f with respect to d :

$$\frac{1}{2}(\log N + \log \lfloor \alpha \log N \rfloor - 2 \log d - 2)$$

This derivative is a decreasing function of d ; its minimum is reached when $d = \lfloor \alpha \log N \rfloor$ where it is equal to:

$$\frac{1}{2}(\log N - \log \lfloor \alpha \log N \rfloor - 2)$$

which is positive for N large enough. Thus $f(w, d)$ is maximum when $w = d = \lfloor \alpha \log N \rfloor$. The maximum of f in the domain is then equal to:

$$\begin{aligned} &\lfloor \alpha \log N \rfloor (\log a + 1/2(\log \lfloor \alpha \log N \rfloor - \log N)) + \frac{\lfloor \alpha \log N \rfloor}{2} (\log N - \log \lfloor \alpha \log N \rfloor) \\ &= \lfloor \alpha \log N \rfloor \log a = O(1) \log N \end{aligned}$$

Thus by inequality 1, we get for all (w, d) in the domain:

$$f(w, d) \leq O(1) \log N + \log r (w - \lfloor \alpha \log N \rfloor)$$

This proves in turn that :

$$p_N(w, d) \leq N^{-\frac{dc}{2}+1} N^{\frac{O(1)}{\log(\log N)}} r^{w-\lfloor \alpha \log N \rfloor}$$

Thus for each $d \in \llbracket 0; \lfloor \alpha \log N \rfloor \rrbracket$, since $r < 1$:

$$\begin{aligned} \sum_{w=\lfloor \alpha \log N \rfloor+1}^{\lfloor xN \rfloor-1} p_N(w, d) &\leq \sum_{w=\lfloor \alpha \log N \rfloor}^{\infty} N^{-\frac{dc}{2}+1} N^{\frac{O(1)}{\log(\log N)}} r^{w-\lfloor \alpha \log N \rfloor} \\ &\leq O(1) N^{-\frac{dc}{2}+1} N^{\frac{O(1)}{\log(\log N)}} \end{aligned}$$

Consequently, by summing over d , the second partial sum is upper bounded by:

$$O(1) (\log N + 1) N^{-\frac{d_c}{2}+1} N^{\frac{O(1)}{\log(\log N)}}$$

which tends to 0 when $N \rightarrow \infty$ since $d_c > 2$. □

Proof of Lemma 6.11 **Third partial sum, sub-log case**

Proof. The proof is exactly the same as the third partial sum in the polynomial case, by changing every occurrence of N^α by $\alpha \log N$. □

References

- [1] N. Kahale and R. Urbanke On the minimum distance of parallel and serially concatenated codes *Proc. IEEE Int. Symp. Inf. Theory (ISIT'98)*, p.31, 1998.
- [2] D. Poulin and J.-P. Tillich and H. Ollivier Quantum serial turbo-codes *IEEE Trans. Inf. Theory*, 55(6):2776–2798, 2009.